

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
Санкт-Петербургский горный университет

**Кафедра информатики и компьютерных технологий**

**ИНФОРМАТИКА**  
**ПРОГРАММИРОВАНИЕ В VBA**  
Процедуры и функции  
Методические указания к лабораторным работам

САНКТ-ПЕТЕРБУРГ  
2018

УДК 004.67(076)

ИНФОРМАТИКА. ПРОГРАММИРОВАНИЕ В VBA. Процедуры и функции. Методические указания к лабораторным работам / Санкт-Петербургский горный университет. Сост.: Г.Н.Журов, Л.Г.Муста, СПб, 2018, 19 с.

Целью предлагаемых работ является изучение основ работы с Visual Basic for Applications (VBA) в Excel и освоение на практике использования подпрограмм на языке программирования VBA.

Методические указания предназначены для студентов всех направлений.

Научный редактор доц. *А.Б. Маховиков*

Табл. 3. Ил. 6. Библиогр. 3.

© Санкт-Петербургский горный университет, 2018

## **ВВЕДЕНИЕ**

В Государственный образовательный стандарт высшего профессионального образования по дисциплине «Информатика» включено изучение основ программирования на языке высокого уровня, технологий программирования. Visual Basic for Applications (VBA) в Excel – это самый современный язык программирования со средой программирования, которая запускается из Excel как отдельное приложение. Программа Excel берет на себя большую часть работы по организации проекта (программы пользователя), избавляя преподавателя от объяснения многих технических мелочей. Практически сразу студенты могут начинать писать программу.

Другое достоинство программирования на VBA в Excel – студенты имеют дело с одной программой, табличным процессором, как центром обработки данных. Наличие надстроек (библиотек программ) позволяет организовать сложную обработку данных. Умение программировать на VBA позволяет дополнить имеющиеся надстройки собственными программами.

VBA — упрощённая реализация языка программирования Visual Basic, встроенная в линейку продуктов Microsoft Office, а также во многие другие программные пакеты, такие как AutoCAD, SolidWorks, CorelDRAW. VBA является интерпретируемым языком. Как и следует из его названия, VBA близок к Visual Basic. В будущем Microsoft планирует заменить VBA на Visual Studio Tools for Applications (VSTA) — инструментарий расширения функциональности приложений, основанный на Microsoft .NET.

Методические указания содержат краткие теоретические сведения, необходимые для работы в редакторе VBA и позволяющие составлять программы с использованием подпрограмм-функций и подпрограмм-процедур. Приведены варианты заданий и рассмотрен пример решения одного варианта.

Полученные в ходе выполнения лабораторной работы навыки позволят в дальнейшем использовать их в курсовых и дипломных работах, а также в практической деятельности.

## ЛАБОРАТОРНАЯ РАБОТА

### ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ПОДПРОГРАММ

Цель работы - освоить на практике применение подпрограмм-процедур и подпрограмм-функций.

#### ОБЩИЕ СВЕДЕНИЯ

*Подпрограмма* — программа, реализующая вспомогательный алгоритм.

*Основная программа* — программа, реализующая основной алгоритм решения задачи и содержащая в себе обращения к подпрограммам.

В VBA существуют два типа подпрограмм: подпрограммы-функции и подпрограммы-процедуры.

Отличие функции от процедуры заключается в том, что результатом исполнения операторов, образующих тело функции, всегда является некоторое единственное значение, поэтому обращение к функции можно использовать в соответствующих выражениях наряду с переменными и константами.

Структурным элементом программы, написанной на языке VBA, является модуль – совокупность объявлений, процедур и функций, объединенных в единое целое.

Каждый модуль состоит из области описания и одной или нескольких процедур и функций.

Входящие в модуль процедуры и функции объединены общей областью описания. В ней описываются данные и объекты, которые являются общепринятыми для подпрограмм модуля.

*Подпрограмма-процедура* имеет следующий синтаксис:

**[Private | Public] [Static] Sub <имя процедуры> ([<список аргументов>])**

**[<Инструкции>]**

**[Exit Sub]**

**[<Инструкции>]**

**End Sub**

**Public** – указывает, что процедура Sub доступна во всех других процедурах во всех модулях;

**Private** – указывает, что процедура Sub доступна для других процедур только того модуля, в котором она описана;

**Static** – указывает, что локальные переменные процедуры Sub сохраняются в промежутках времени между вызовами этой процедуры;

**Sub, End Sub** – служебные слова VBA;

- **<имя процедуры>** – имя процедуры, удовлетворяющее стандартным правилам именования;

- **<список аргументов>** – список переменных, представляющих аргументы, которые передаются в процедуру Sub при её вызове. Имена переменных разделяются запятой;

- **<Инструкции>** – любой набор команд VBA;

- **Exit Sub** – инструкция, выполнение которой приводит к выходу из процедуры.

В качестве результата процедура может возвращать в вызывающую программу множество простых или структурированных величин или не возвращать никаких значений.

Среди параметров процедуры указываются как аргументы, так и результаты.

Обращение к процедуре — отдельный оператор.

Вызов процедуры из другой процедуры можно произвести несколькими способами.

Первый способ:

**<Имя процедуры> <Список фактических параметров>**

- **<Имя процедуры>** – имя вызываемой процедуры;
- **<Список фактических параметров>** – список аргументов, передаваемых процедуре; он должен соответствовать списку, заданному в процедуре, по количеству и типу.

Второй способ:

**Call <Имя процедуры> (<Список фактических параметров>)**

- **Call** – служебное слово VBA;

- **<Имя процедуры>** – имя вызываемой процедуры;
- **<Список фактических параметров>** – список аргументов, передаваемых процедуре; он должен соответствовать списку, заданному в процедуре, по количеству и типу.

Заметим, что при втором способе вызова процедуры в отличие от первого список фактических параметров должен быть заключен в круглые скобки; в качестве разделителя в списке

*Подпрограмма-функция* имеет следующий синтаксис:

```
[Private | Public] [Static] Function <имя функции>  
[(<список аргументов>)]  
    [<Инструкции>]  
    [Exit Function]  
    [<Инструкции>]  
End Sub
```

Синтаксис инструкции **Function** содержит те же элементы, что и **Sub**. Инструкция **Exit Function** приводит к выходу из функции.

Тип функции может быть только простым типом.

Блок содержит локальные для функции описания и раздел операторов.

Для возврата значения из функции следует присвоить значение имени функции.

Обращение к функции является операндом в выражении.

Подпрограмма-функция вызывается в выражении по своему имени, за которым следует вписок аргументов в скобках

Для работы подпрограммам иногда необходимо передать данные, которые носят названия параметров или аргументов подпрограммы. Параметры передаются в подпрограммы при ее вызове при помощи перечисления их после имени подпрограммы. При этом описание действий в подпрограмме осуществляется с использованием *формальных параметров*. Обращение к подпрограмме осуществляется с *фактическими параметрами*, которые должны соответствовать формальным по числу, типу и месту расположения.

Использование подпрограммы Function аналогично использованию стандартных функций. Обращение к ней можно записать , например, в правой части оператора присваивания, при этом указывается имя подпрограммы- функции и в круглых скобках – фактические параметры т.е. *список аргументов*, заменяющих формальные параметры в операторе Function

## ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

### Задача 1.

Заданы два вектора А и В. Вычислить сумму элементов для каждого вектора. Вычисление суммы элементов оформить в виде функции.

### Решение:

На листе MS Excel разместим исходные векторы :

	A	B	C	D	E	F	G
1		A					
2	2,1	5,4	9,4				
3		B					
4	5,1	-2,5	4,8	-5,6	3,2		

Рис. 1

В данном примере требуется дважды вычислить сумму элементов вектора. Результатом вычисления в каждом случае будет одно число, поэтому целесообразно организовать подпрограмму-функцию.

Формальными параметрами можно выбрать X и n, где X – исходный вектор, n – его размерность.

### Текст программы:

```
Option Base 1
Option Explicit
Dim i As Integer
```

### 'Подпрограмма-функция

```
Public Function Sum(X() As Single, n As Integer) As Single
' Вычисление суммы элементов вектора X размерности n
For i = 1 To n
Sum = Sum + X(i)
Next
End Function
```



### 'Основная программа

```
Public Sub Primer1()  
Dim A(3) As Single, B(5) As Single, SA As Single, SB As Single  
'Чтение элементов исходных векторов с листа MS Excel  
For i = 1 To 3  
A(i) = Cells(2, i)  
Next  
For i = 1 To 5  
B(i) = Cells(4, i)  
Next  
'Обращение к процедуре-функции  
SA = Sum(A, 3)  
SB = Sum(B, 5)  
'Вывод результатов на лист MS Excel  
Cells(1,5)="SA"  
Cells(3,7)="SB"  
Cells(2, 5) = SA  
Cells(4, 7) = SB  
End Sub
```

### Результат работы программы:

	A	B	C	D	E	F	G
1		A			SA		
2	2,1	5,4	9,4		16,9		
3		B					SB
4	5,1	-2,5	4,8	-5,6	3,2		5

Рис. 2

### Задача 2.

Задан вектор  $a$  размерности 3, вектор  $b$  размерности 4. Построить вектор с размерности 3, у которого каждый элемент есть квадрат соответствующего вектора  $a$ . Построить вектор  $d$

размерности 4, у которого каждый элемент есть квадрат соответствующего элемента вектора b.

**Решение.**

В данном примере требуется дважды вычислить новый вектор по известному старому, поэтому целесообразно организовать подпрограмму-процедуру.

Формальными параметрами можно выбрать x, n, y, где x – исходный вектор, n – его размерность, y – новый вектор.

Чтение вектора с листа MS Excel и запись нового вектора на лист также опишем с помощью подпрограмм-процедур.

На листе MS Excel разместим исходные векторы :

	A	B	C	D
1	a			
2	2,5	1	-4	
3	b			
4	-1,5	0	-3,5	2

Рис.3

**Текст программы:**

```
Option Explicit
Option Base 1
Dim i As Integer
```

**'Подпрограмма-функция для формирования нового вектора**

```
Public Sub NewVect(x() As Single, n As Integer, y() As Single)
For i = 1 To n
y(i) = x(i) ^ 2
Next
End Sub
```

**'Подпрограмма-функция для чтения вектора из k – ой строки листа MS Excel**

```
Public Sub VVod(x() As Single, n As Integer, k As Integer)
For i = 1 To n
x(i) = Cells(k, i)
Next
End Sub
```

**'Подпрограмма-функция для записи вектора в к – ую строку листа MS Excel**

```
Public Sub Vyvod(x() As Single, n As Integer, k As Integer)
For i = 1 To n
Cells(k, i) = x(i)
Next
End Sub
```

**'Основная программа**

```
Public Sub Prog2()
Dim a(3) As Single, b(4) As Single
Dim c(3) As Single, D(4) As Single
Call VVod(a, 3, 2)
Call VVod(b, 4, 4)
Call NewVect(a, 3, c)
Call NewVect(b, 4, D)
Cells(5, 1) = "c"
Call Vyvod(c, 3, 6)
Cells(7, 1) = "d"
Call Vyvod(D, 4, 8)
End Sub
```

**Результат работы программы:**

	A	B	C	D
1 a				
2	2,5	1	-4	
3 b				
4	-1,5	0	-3,5	2
5 c				
6	6,25	1	16	
7 d				
8	2,25	0	12,25	4

Рис. 4

### Задача 3.

Найти значение матричного выражения

$$G_{33} = A_{32} * B_{22} * C_{23}, \text{ где } A = \begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 1 & 3 \end{pmatrix}, B = \begin{pmatrix} 4 & 2 \\ 3 & 1 \end{pmatrix}, C = \begin{pmatrix} 2 & 1 & 2 \\ 3 & 0 & 1 \end{pmatrix}$$

### Решение:

Поскольку в данном примере дважды используется умножение матриц, будет рационально написать процедуру для умножения двух прямоугольных матриц произвольной размерности. Назовем эту процедуру Mult, в качестве формальных параметров целесообразно взять следующие: x,y,m,i,n,z, где x и y исходные матрицы размерности m, l и l, n; z – матрица результат размерности m, n.

Главная программа должна содержать ввод исходных матриц A, B, C, двукратное обращение к процедуре Mult и вывод результата.

	A	B	C	D	E	F	G	H	I
1		a			b		c		
2	1	1		4	2		2	1	2
3	2	2		3	1		3	0	1
4	1	3							

Рис.5

**Текст программы:**

```
Option Explicit
Option Base 1
```

```
Dim i As Integer, j As Integer, k As Integer
```

```
Public Sub vvodMatr(x() As Single, m As Integer, n As Integer, _
k1 As Integer, k2 As Integer)
For i = 1 To m
For j = 1 To n
x(i, j) = Cells(i + k1, j + k2)
Next
Next
End Sub
```

```
Public Sub vyvodMatr(x() As Single, m As Integer, n As Integer, _
k1 As Integer, k2 As Integer)
For i = 1 To m
For j = 1 To n
Cells(i + k1, j + k2) = x(i, j)
Next
Next
End Sub
```

```
Public Sub mult(x() As Single, y() As Single, m As Integer, l As Integer,
n As Integer, _
z() As Single)
For i = 1 To m
```

```

For j = 1 To n
For k = 1 To l
z(i, j) = z(i, j) + x(i, k) * y(k, j)
Next
Next
Next
End Sub

```

```

Public Sub Prog3()
Dim a(3, 2) As Single, b(2, 2) As Single, c(2, 3) As Single, _
D(3, 2) As Single, g(3, 3) As Single
Call vvodMatr(a, 3, 2, 1, 0)
Call vvodMatr(b, 2, 2, 1, 3)
Call vvodMatr(c, 2, 3, 1, 6)
Call mult(a, b, 3, 2, 2, D)
Call mult(D, c, 3, 2, 3, g)
Cells(5, 2) = "Результат решения"
Call vyvodMatr(g, 3, 3, 5, 0)
End Sub

```

**Результат работы программы:**

	A	B	C	D	E	F	G	H	I
1		a			b		c		
2	1	1		4	2		2	1	2
3	2	2		3	1		3	0	1
4	1	3							
5		Результат вычисления							
6	23	7	17						
7	46	14	34						
8	41	13	31						

Рис.6

## ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

### Задача 1.

Найти и вывести на печать для исходных векторов соответствующие величины с использованием подпрограммы - функции, где

$$A = \begin{pmatrix} 2.1 \\ -5.6 \\ 8.4 \\ 1.3 \\ 0 \\ 5.8 \end{pmatrix} \quad B = \begin{pmatrix} 1.2 \\ -3.1 \\ 8 \\ -4.5 \\ 0 \\ -2.6 \\ 8.4 \\ 3.2 \\ 1.8 \end{pmatrix} \quad C = \begin{pmatrix} -0.5 \\ 5.9 \\ -2 \\ 0.5 \end{pmatrix} \quad D = \begin{pmatrix} -0.2 \\ -0.5 \\ 9.2 \\ 1.5 \\ 0.4 \\ -2 \\ -5.6 \\ 4.8 \\ 9 \\ 4.6 \\ 3 \end{pmatrix}$$

$$F = \begin{pmatrix} 0.2 \\ -8.5 \\ 4.5 \\ 5.6 \\ -4 \end{pmatrix} \quad G = \begin{pmatrix} -8.1 \\ 0.3 \\ 0 \\ -1.2 \\ 0 \\ 2.4 \\ 3 \end{pmatrix}$$

Таблица 1

Вариант	Данные вектора	Вычисляемые величины для каждого вектора
1	A, B	Сумма квадратов элементов
2	F, G	Сумма модулей элементов
3	A, C	Минимальный элемент
4	B, D	Максимальный элемент
5	F, A	Сумма элементов, больших единицы
6	F, B	Число неотрицательных элементов
7	G, D	Произведение всех элементов
8	G, A	Произведение всех отрицательных элементов
9	A, D	Произведение всех элементов, больших двух
10	A, C	Число отрицательных элементов
11	B, D	Сумма элементов, больших 0,5
12	D, F	Число ненулевых элементов
13	D, C	Сумма положительных элементов
14	G, D	Сумма отрицательных элементов
15	A, F	Произведение положительных элементов

**Задача 2.**

По данным таблицы 2 найти и вывести на печать два вектора. Исходные векторы те же, что и задаче 1.

Таблица 2

Вариант	Исходные векторы	Вычисляемые векторы	Элементы вычисляемых векторов по отношению к исходным
1	A, B	X, Y	Модуль элемента
2	F, G	Y, Z	На 1 больше
3	A, C	P, T	На 3 меньше
4	B, D	T, V	В 5 раз меньше
5	F, A	V, W	В 10 раз больше
6	F, B	N, X	Куб элемента
7	G, D	Y, V	Модуль элемента, увеличенный на 1



Вариант	Исходные векторы	Вычисляемые векторы	Элементы вычисляемых векторов по отношению к исходным
8	G, A	V,Z	Синус элемента
9	A, D	V,R	Косинус элемента
10	A, C	R,X	Тангенс элемента
11	B, D	X,R	На 3 меньше
12	D, F	Y,R	В 5 раз больше
13	D, C	W,X	Куб элемента, увеличенный на 2
14	G, D	W,T	Модуль элемента
15	A, F	W,T	В 3 раза больше

### Задача 3.

По данным таблицы 3 вычислить и вывести на печать два вектора. Вычисление всех искомым величин оформить в виде подпрограмм-процедур.

Исходные матрицы:

$$A_{23} = \begin{pmatrix} 0.1 & -0.1 & 0.5 \\ 0.2 & -0.2 & 0.7 \end{pmatrix} \quad B_{22} = \begin{pmatrix} -1 & 0.1 \\ -0.1 & 2 \end{pmatrix} \quad C_{33} = \begin{pmatrix} 0.1 & 0.2 & 0.3 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}$$

$$D_{32} = \begin{pmatrix} -1.1 & 0 \\ 2.1 & 0 \\ 0 & -7.3 \end{pmatrix} \quad E_{22} = \begin{pmatrix} 1 & -4.3 \\ -2.7 & 0 \end{pmatrix}$$

Таблица 3.

Вариант	Данные матрицы	Искомые векторы	Значения элементов векторов
1	A <sub>23</sub> , B <sub>22</sub>	X <sub>2</sub> , Y <sub>2</sub>	$X_i = \sum_{j=1}^3 A_{i,j}, Y_i = \sum_{j=1}^2 B_{i,j}$

Вариант	Данные матрицы	Искомые векторы	Значения элементов векторов
2	$C_{33}, D_{32}$	$U_3, V_2$	$U_j = \sum_{i=1}^3 \sin C_{i,j}, V_j = \sum_{i=1}^3 \sin D_{i,j}$
3	$D_{32}, F_{22}$	$U_3, V_2$	$U_i = \sum_{j=1}^2 \cos D_{i,j}, V_j = \sum_{i=1}^2 \cos E_{i,j}$
4	$A_{23}, D_{32}$	$U_3, V_2$	$U_j = \sum_{i=1}^2 A^2_{i,j}, V_j = \sum_{i=1}^3 D^2_{i,j}$
5	$A_{23}, C_{33}$	$V_2, U_3$	$V_i = \sum_{j=1}^3 (A_{i,j} - 2), U_i = \sum_{j=1}^3 (C_{i,j} - 2)$
6	$C_{33}, E_{22}$	$U_3, Y_2$	$U_i = \sum_{j=1}^3 (C_{i,j} - 3), Y_i = \sum_{j=1}^3 (E_{i,j} - 3)$
7	$E_{22}, B_{22}$	$X_2, Y_2$	$X_i = \sum_{j=1}^2 (-1)^{i+j} E_{i,j}, Y_i = \sum_{j=1}^2 (-1)^{i+j} B_{i,j}$
8	$B_{22}, C_{33}$	$X_2, U_3$	$X_i = \sum_{j=1}^2 \operatorname{tg} B_{i,j}, U_i = \sum_{j=1}^3 \operatorname{tg} C_{i,j}$
9	$C_{33}, E_{22}$	$U_3, Y_2$	$U_j = \sum_{i=1}^3 \operatorname{tg} B_{i,j}, Y_j = \sum_{i=1}^2 \operatorname{tg} E_{i,j}$
10	$E_{22}, D_{32}$	$X_2, Y_3$	$X_i = \sum_{j=1}^2  E_{i,j} , Y_i = \sum_{j=1}^2  D_{i,j} $
11	$D_{32}, E_{22}$	$U_2, V_2$	$U_j = \sum_{i=1}^3 D^2_{i,j}, V_j = \sum_{i=1}^2 E^2_{i,j}$
12	$D_{32}, B_{22}$	$U_3, V_2$	$U_i = \sum_{j=1}^2 (D_{i,j} - 1)^2, V_i = \sum_{j=1}^2 (B_{i,j} - 1)^2$
13	$A_{23}, C_{33}$	$U_3, V_3$	$U_j = \sum_{i=1}^2 (A_{i,j} - 2)^2, V_j = \sum_{i=1}^3 (C_{i,j} - 2)^2$

Вариант	Данные матрицы	Искомые векторы	Значения элементов векторов
14	$A_{23}, B_{22}$	$X_2, Y_2$	$X_i = \sum_{j=1}^3 A^3_{i,j}, Y_i = \sum_{j=1}^2 B^3_{i,j}$
15	$C_{33}, D_{32}$	$U_3, V_2$	$U_j = \sum_{i=1}^3 (C_{i,j} + 2)^2, V_j = \sum_{i=1}^3 (D_{i,j} + 2)^2$

### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Информатика. Базовый курс: учебное пособие / под ред. С.В. Симоновича. — СПб. и др.: Питер, 2012. — 637 с.
2. Информатика: Учебник для вузов / Под ред. Н.В. Макаровой – 3-е изд., перераб. - М.: Финансы и статистика, 2009. - 768 с.
3. Веденева Е.А. Функции и формулы Excel 2007. Библиотека пользователя. –СПб.: Питер, 2008. –384 с

### СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
ОБЩИЕ СВЕДЕНИЯ .....	4
ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ .....	8
ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ .....	15