

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

**Кафедра информатики и компьютерных технологий**

**ИНФОРМАТИКА**  
**ОРГАНИЗАЦИЯ ЦИКЛИЧЕСКИХ**  
**ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ В VBA**

*Методические указания к лабораторным работам  
для студентов специальностей 21.05.02, 21.05.04*

**САНКТ-ПЕТЕРБУРГ**  
**2020**

УДК 004.424.22 (073)

**ИНФОРМАТИКА: Организация циклических вычислительных процессов в VBA:** Методические указания к лабораторным работам / Санкт-Петербургский горный университет. Сост.: *Е.Н. Овчинникова, В.Н. Сибирев*. СПб, 2020. 36 с.

В методических указаниях к лабораторным работам изложен теоретический материал и приведены примеры выполнения заданий, необходимых для решения циклических задач в среде программирования Visual BASIC for Applications. В конце каждой работы даны варианты для индивидуальных заданий.

Предназначены для студентов специальностей 21.05.02 «Прикладная геология» и 21.05.04 «Горное дело» (специализация «Шахтное и подземное строительство»).

Научный редактор доц. *А.Б. Маховиков*

Рецензент проф. *С.М. Одоевский* (Санкт-Петербургская военная академия связи)

## **ВВЕДЕНИЕ**

Visual Basic for Applications (VBA) – это упрощённая версия среды программирования Visual Basic, встроенная в линейку продуктов Microsoft Office, а также во многие другие программные пакеты, в том числе AutoCAD, CorelDraw, Adobe Creative Suite и WordPerfect. В среде VBA объединились возможности языков программирования высокого уровня и средства визуального и объектно-ориентированного программирования.

VBA является полноценной средой программирования, позволяющей записать не только последовательно выполняемые пользователем действия, но и конструкции, включающие разнообразные средства для организации ветвлений и циклов.

В методических указаниях рассмотрены основные возможности среды VBA для решения циклических задач. Представлены три лабораторные работы, включающие приемы вычислений с использованием операторов цикла, табулирование функций, технологию обработки одномерных и двумерных массивов.

В начале каждой лабораторной работы приводятся необходимые теоретические сведения по теме работы. Затем разбираются учебные примеры, которые служат образцом для выполнения индивидуальных заданий.

Цель методических указаний: содействовать формированию общекультурных и общепрофессиональных компетенций (умение пользоваться компьютером как средством управления и обработки информационных массивов; способность решать задачи профессиональной деятельности на основе информационной культуры с применением информационно-коммуникационных технологий); способствовать развитию внимательности, целеустремленности, навыков самоорганизации и самоконтроля.

Учебно-методический материал может быть использован как на аудиторных лабораторных занятиях, так и в рамках самостоятельной работы студентов специальностей 21.05.02 «Прикладная геология» и 21.05.04 «Горное дело» (специализация «Шахтное и подземное строительство»).

# ЛАБОРАТОРНАЯ РАБОТА № 1

## РЕШЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ С ПРИМЕНЕНИЕМ ОПЕРАТОРОВ ЦИКЛА

### 1.1. ОБЩИЕ СВЕДЕНИЯ

При решении многих задач возникает необходимость многократного повторения одних и тех же действий, но при различных значениях переменных, определяющих эти действия. Такие вычислительные процессы называются *циклическими*, а многократно повторяемые участки – *циклами*.

Типичным примером циклического процесса является задача *табулирования функции*: чтобы получить таблицу значений некоторой функции на заданном интервале изменения аргумента с заданным шагом, необходимо соответствующее количество раз определить следующее значение аргумента и вычислить для него значение функции.

Любой алгоритм циклической структуры в общем случае содержит следующие действия:

- 1) задание начальных значений переменным, изменяющимся в цикле (подготовка цикла);
- 2) действия, выполняемые непосредственно в цикле, называемые *телом цикла*;
- 3) изменение значений переменных цикла по некоторому правилу;
- 4) проверка условия продолжения или окончания цикла и, соответственно, переход к началу цикла, если он не завершен, или выход из цикла по его окончании.

Процессы вычислений циклической структуры можно разделить на две группы:

1. Циклические процессы, для которых количество повторений известно (циклы с заданным числом повторений).
2. Циклические процессы, завершающиеся по достижении или нарушении некоторых условий.

В соответствии с данной классификацией, выделяют следующие разновидности циклов:

- *цикл со счетчиком* (цикл с параметром) – цикл с известным числом повторений;

- *цикл с предусловием* – действия внутри этого цикла повторяются, пока выполняется некоторое условие в блоке ветвления; причем сначала проверяется условие, а затем выполняется действие;

- *цикл с постусловием* – проверка условия выхода из цикла выполняется после очередного циклического действия.

В среде программирования VBA для реализации циклических вычислительных действий используются три оператора цикла **For...Next**, **While...Wend**, **Do... Loop**.

### Оператор цикла **For...Next**

Для выполнения циклических процессов с заранее известным числом повторений применяется оператор цикла **For...Next**, который реализует цикл с заранее известным числом повторений:

**For** *Имя Переменной* = *Начало Цикла* **To** *Конец Цикла* **Step**  
*Шаг Цикла*

*Операторы Тела Цикла*

**Next**

Работа цикла происходит следующим образом:

- 1) переменной цикла присваивается начальное значение *Начало Цикла*;

- 2) выполняются *Операторы Тела Цикла*;

- 3) оператор **Next** возвращает управление оператору **For**.

Оператор **For** увеличивает значение переменной цикла на шаг и проверяет условие окончания цикла: пока значение переменной цикла меньше или равно конечному значению, выполняется очередная итерация цикла; в противном случае происходит выход из цикла.

Если *Шаг Цикла* = 1, то оператор **Step** опускают.

Фрагмент блок-схемы, который реализует оператор цикла **For...Next**, можно представить следующим образом (рис. 1.1):

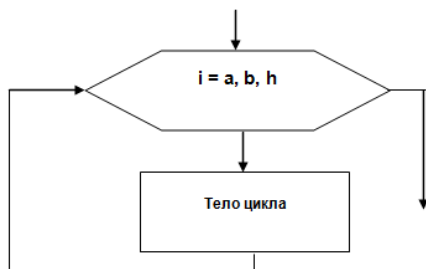


Рис. 1.1. Графическая структура цикла с заранее известным числом повторений

### Оператор цикла **While...Wend**

Оператор цикла **While...Wend** используется в том случае, когда число повторений операторов тела цикла заранее неизвестно, и циклические действия продолжаются в зависимости от некоторого условия.

Синтаксис оператора:

```

While <условие>
    <операторы цикла>
Wend
  
```

Оператор **While...Wend** работает следующим образом: проверяется некоторое условие. Если оно выполняется, то работают операторы тела цикла. Если же условие не выполняется – происходит переход к следующему оператору после служебного слова **Wend**.

Следовательно, оператор цикла **While...Wend** реализует *цикл с предусловием*. Графическая структура цикла с предусловием представлена на рис. 1.2.



Рис. 1.2. Графическая структура цикла с предусловием

### Оператор цикла с условием **Do ... Loop**

Оператор цикла **Do...Loop** применяется в том случае, когда число повторений операторов тела цикла заранее неизвестно. С помощью оператора **Do...Loop** в среде VBA можно реализовать *цикл с предусловием* и *цикл с постусловием*.

Формат *цикла с предусловием*:

**Do While** <условие>  
           <операторы цикла>

#### **Loop**

Перед каждым выполнением цикла проверяется условие. Если условие истинно, то происходит выполнение операторов цикла.

Если же условие ложно, то происходит выход из цикла и переход к следующему оператору программы. Т.е., цикл с предусловием никогда не выполняется в случае невыполнения условия.

Формат *цикла с постусловием*:

**Do**  
           <операторы цикла (тело цикла)>

**Loop While/Until** <условие>

Цикл с постусловием – это цикл, в котором условие проверяется после выполнения тела цикла. Следовательно, операторы цикла всегда выполняются хотя бы один раз, независимо от истинности условия. Графическая структура *цикла с постусловием* приведена на рис. 1.3.

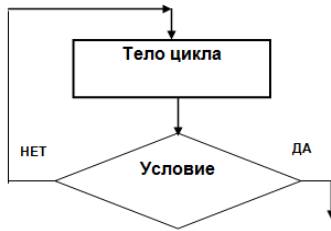


Рис. 1.3. Графическая структура цикла с постусловием

**Примечание.** Проверка условий выхода из цикла проводится с помощью ключевых слов **While** или **Until**. Эти слова придают одному и тому же условию противоположный смысл.

Например, в цикле:

$A = 1$

**Do**

$A = A + 1$

Print "Visual Basic"

**Loop While**  $A < 11$

- условие выхода из цикла можно трактовать следующим образом: *пока  $A$  меньше 11*. В данном случае фраза «Visual Basic» будет печататься, пока значение переменной  $A$  меньше одиннадцати (т. е. десять раз).

В цикле:

$A = 1$

**Do**

$A = A + 1$

Print "Visual Basic"

**Loop Until**  $A > 10$

- условие выхода из цикла можно трактовать так: *до тех пор, пока значение переменной  $A$  не стало больше десяти*. В данном случае фраза «Visual Basic» будет распечатываться также 10 раз. Таким образом, если в цикле с постусловием применяется ключевое слово **Until**, то цикл повторяется при ложном значении условия. Если же проверяемое условие стало истинным, цикл заканчивается.



## 1.2. ПРИМЕРЫ РЕШЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ

**Пример 1.** Вычислить сумму:  $S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{12}$

### Анализ задачи

Вычисление конечной суммы сводится к нахождению суммы заданного количества слагаемых:

$$S = f(1) + f(2) + \dots + f(n) = \sum_{i=1}^n f(i), \quad (1.1)$$

где  $i$  – номер слагаемого;  $f(i)$  – слагаемое с номером  $i$ .

Вычисление суммы в данном случае организуется в виде циклического процесса, когда при каждом прохождении цикла номер слагаемого  $i$  увеличивается на единицу, а сумма изменяется на величину  $i$ -го слагаемого. Цикл будет продолжаться до тех пор, пока не будут просуммированы все  $n$  слагаемых.

Введем обозначения:

$i$  – переменная цикла;

$s$  – сумма слагаемых;

В данном примере переменная цикла  $i$  изменяется от 1 до 12 с шагом 1. Следовательно, значение результата можно записать в виде формулы:

$$S = \sum_{i=1}^{12} \frac{1}{i} \quad (1.2)$$

Рекуррентное соотношение для суммирования запишется следующим образом:

$$S = S + \frac{1}{i}, \quad (1.3)$$

что означает добавление слагаемого  $f(i)=1/i$  с номером  $i$  к значению суммы, вычисленному на предыдущем шаге алгоритма, и присваивание вычисленного значения  $S + f(i)$  той же переменной  $S$ .

Поскольку в данном примере шаг изменения переменной цикла равен 1, то для вычисления конечной суммы можно использовать любой из трех рассмотренных операторов цикла: **For...Next**, **Do... Loop** или **While...Wend**.

## Решение

1. Оформить лист Excel согласно условию задачи (рис. 1.4).

	A	B	C	D	E
1	Пример 1.				
2	Вычислить сумму чисел:			$S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{12}$	
3					
4		сумма =			

Рис. 1.4. Оформление листа Excel для решения примера 1

2. Вызвать редактор VBA, создать новый модуль с именем «Пример\_1»; набрать программный код с применением оператора цикла **For...Next**:

```
Public Sub пример_1()  
Dim i As Integer  
Dim s As Single  
s = 0 'задание начального значения суммы  
For i = 1 To 12 'задание начального и конечного значений  
переменной цикла; шаг не указан, т.к. равен 1  
s = s + 1 / i 'оператор накопления суммы  
Next i 'оператор продолжения цикла  
Range("C4") = s 'запись результата в ячейку C4  
End Sub
```

3. Запустить программу на выполнение. Убедиться, что в ячейке **C4** появился результат вычислений.

4. Решить эту же задачу с применением оператора цикла **While...Wend** (создать новый модуль). Ответ записать в ячейку **C5**:

```
Public Sub пример_11()  
Dim i As Integer  
Dim s As Single  
s = 0  
i = 1 'начальное значение переменной цикла  
While i <= 12 'проверка условия продолжения цикла
```

```

s = s + 1 / i 'оператор накопления суммы
i = i + 1 'изменение значения переменной цикла
Wend
Range("C5") = s 'запись результата в ячейку C5
End Sub

```

5. Решить эту же задачу с применением оператора цикла **Do...Loop**. Ответ записать в ячейку C6:

```

Public Sub Пример_12()
Dim i As Integer
Dim s As Single
s = 0
i = 1
Do
    s = s + 1 / i
    i = i + 1
Loop Until i > 12 'проверка условия продолжения цикла
Range("C6") = s 'запись результата в ячейку C6
End Sub

```

6. Убедиться, что результаты вычислений, полученные разными способами, совпали (рис. 1.5).

	A	B	C	D	E
1	Пример 1.				
2	Вычислить сумму чисел:			$S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{12}$	
3					
4		сумма =	3,103211		
5			3,103211		
6			3,103211		

Рис. 1.5. Результаты решения примера 1 разными способами

**Пример 2.** Вычислить сумму:  $S = \frac{1}{3} + \frac{1}{5} + \dots + \frac{1}{15}$

### Анализ задачи

Решение задачи аналогично примеру 1. В данном примере переменная цикла  $i$  изменяется от 3 до 15 с шагом 2. Для вычисления конечной суммы можно также использовать все три операторы цикла.

### Решение

1. Оформить лист Excel согласно условию задачи (рис. 1.6).

	A	B	C	D	E
1	Пример 2.				
2	Вычислить сумму чисел: $S = \frac{1}{3} + \frac{1}{5} + \dots + \frac{1}{15}$				
3					
4		сумма =			

Рис. 1.6. Оформление листа Excel для решения примера 2

2. Вызвать редактор VBA, создать новый модуль с именем «Пример\_2»; набрать программный код с применением оператора цикла **Do...Loop**:

```

Public Sub Пример_2()
Dim i As Integer
Dim s As Single
s = 0 'задание начального значения суммы
i = 3 'начальное значение переменной цикла
Do
    s = s + 1 / i 'оператор накопления суммы
    i = i + 2 'изменение значения переменной цикла
Loop While i <= 15 'проверка условия продолжения цикла
Range("C4") = s 'запись результата в ячейку C4
End Sub

```

3. Запустить программу на выполнение. Убедиться, что в ячейке **C4** появился результат вычислений (рис. 1.7).

	A	B	C	D	E
1	<b>Пример 2.</b>				
2	<b>Вычислить сумму чисел:</b>		$S = \frac{1}{3} + \frac{1}{5} + \dots + \frac{1}{15}$		
3					
4		<b>сумма =</b>	1,0218		
5					

Рис. 1.7. Результат решения примера 2

4. Решить эту же задачу, организовав вывод результата вычислений в диалоговое окно:

**Public Sub Пример\_22()**

**Dim i As Integer**

**Dim s As Single**

**s = 0** 'задание начального значения суммы

**i = 3** 'начальное значение переменной цикла

**Do**

**s = s + 1 / i** 'оператор накопления суммы

**i = i + 2** 'изменение значения переменной цикла

**Loop While i <= 15** 'проверка условия продолжения цикла

**MsgBox ("сумма=" + Str(s))** 'вывод результата в диалоговое

окно

**End Sub**

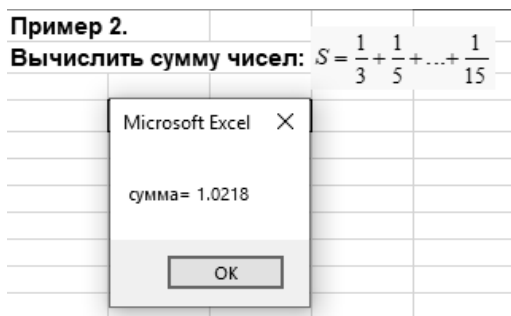


Рис. 1.8. Результат решения примера 2 вторым способом

**Пример 3.** Найти сумму ряда:  $\sum_{n=1}^{100} \frac{1}{n^2}$

Суммирование продолжать до тех пор, пока сумма не достигнет значения 1,6. Определить количество слагаемых.

### Анализ задачи

В данном примере переменная цикла  $i$  изменяется от 1 до 100 с шагом 1. Так как продолжение цикла зависит от условия, для вычисления конечной суммы можно использовать два оператора цикла: **Do... Loop** или **While...Wend**.

### Решение

Программный код для решения задачи:

```
Public Sub Пример_3()
```

```
Dim n As Integer
```

```
Dim S As Single
```

```
n = 1
```

```
S = 0
```

*'Суммирование продолжать, пока сумма не достигнет значения 1,6*

```
Do While S < 1.6
```

```
S = S + 1 / (n * n)
```

```
n = n + 1
```

```
Loop
```

```
MsgBox ("Сумма = "& S & "Число слагаемых = "& n - 1)
```

```
End Sub
```

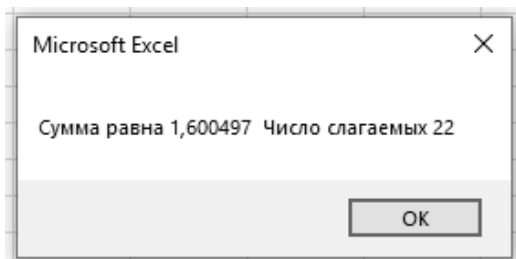


Рис. 1.9. Результат решения примера 3

**Пример 4.** Вычислить сумму:  $S = \sum_{i=1}^7 \frac{i+1}{i!}$

### Анализ задачи

Введем обозначения:

$i$  – переменная цикла;

$s$  – сумма слагаемых;

В данном примере переменная цикла  $i$  изменяется от 1 до 7 с шагом 1.

Для вычисления факториала  $i!$  необходимо ввести дополнительную переменную, например,  $P$ . Вычисление факториала происходит циклически по формуле:

$$P = P * i \quad (1.4)$$

Следовательно, рекуррентное соотношение для суммирования запишется следующим образом:

$$S = S + \frac{i+1}{P} \quad (1.5)$$

Для вычисления конечной суммы можно использовать любой из трех рассмотренных операторов цикла: **For...Next**, **Do...Loop** или **While...Wend**.

### Решение

1. Оформить лист Excel согласно условию задачи (рис. 1.10).

	A	B	C	D
1	<b>Пример 4.</b>			
2	<b>Вычислить сумму:</b>		$S = \sum_{i=1}^7 \frac{i+1}{i!}$	
3				
4		<b>сумма =</b>		
5				

Рис. 1.10. Оформление листа Excel для решения примера 4

2. Вызвать редактор VBA, создать новый модуль с именем «Пример\_4»; набрать программный код с применением оператора цикла **For...Next**:

```

Public Sub Пример_4()
Dim i, p As Integer
Dim s As Single
p = 1 'задание начального значения произведения
s = 0 'задание начального значения суммы
For i = 1 To 7 'задание начального и конечного значений
переменной цикла; шаг не указан, т.к. равен 1
p = p * i 'вычисление факториала
s = s + (i + 1) / p 'оператор накопления суммы
Next i 'оператор продолжения цикла
Range("C4") = s 'запись результата в ячейку C4
End Sub

```

3. Запустить программу на выполнение. Убедиться, что в ячейке **C4** появился результат вычислений (рис. 1.11).

	A	B	C	D
1	<b>Пример 4.</b>			
2	<b>Вычислить сумму:</b> $S = \sum_{i=1}^7 \frac{i+1}{i!}$			
3				
4		<b>сумма =</b>	4,43631	

Рис. 1.11. Результат решения примера 4

### 1.3. ВАРИАНТЫ ЗАДАНИЙ

**Задача 1.** Вычислить сумму чисел, применив операторы цикла **For...Next**, **Do... Loop** и **While...Wend**. Вывод результатов организовать как в ячейки Excel, так и в диалоговые окна.

Определить количество слагаемых, при которых сумма достигнет конкретного значения.



Таблица 1.1

№ вар-та	Выражение для суммирования	Предельная сумма
1	$S= 1 + 1/4 + 1/7 + \dots + 1/25$	1,5
2	$S= 1/2 + 1/4 + 1/6 + \dots + 1/26$	1,3
3	$S= 1/3 + 1/5 + 1/7 + \dots + 1/27$	1,1
4	$S= 1/2 + 1/5 + 1/8 + \dots + 1/35$	1
5	$S= 1/3 + 1/6 + 1/9 + \dots + 1/42$	0,9
6	$S= 1 + 1/5 + 1/9 + \dots + 1/45$	1,4
7	$S= 1/3 + 1/7 + 1/11 + \dots + 1/55$	0,7
8	$S= 1/2 + 1/6 + 1/10 + \dots + 1/54$	0,8
9	$S= 1/3 + 1/8 + 1/13 + \dots + 1/68$	0,5
10	$S= 1/2 + 1/7 + 1/12 + \dots + 1/67$	0,7
11	$S= 1/5 + 1/7 + 1/9 + \dots + 1/31$	0,5
12	$S= 1/4 + 1/7 + 1/10 + \dots + 1/43$	0,8
13	$S= 1/3 + 1/9 + 1/15 + \dots + 1/75$	0,4
14	$S= 1 + 1/6 + 1/11 + \dots + 1/56$	1,2
15	$S= 1/4 + 1/8 + 1/12 + \dots + 1/52$	0,6

**Задача 2.** Вычислить сумму  $\sum_{i=N1}^{N2} f(i)$ , применив операторы цикла **For...Next**, **Do... Loop** и **While...Wend**.

Вывод результатов организовать как в ячейки Excel, так и в диалоговые окна.

Таблица 1.2

№ вар-та	$f(i)$	$NI$	$N2$
1	$\frac{(-1)^i}{(i+1)!}$	1	8
2	$\frac{e^i}{(i+2)!}$	1	7
3	$\frac{i+1}{(i+1)!}$	1	9
4	$\frac{1}{\sqrt{(i+1)!}}$	1	10
5	$\frac{2}{(i+1)!}$	1	7
6	$3^i(i+3)!$	1	6
7	$\frac{(-1)^i}{i!}$	1	6
8	$(i+2)!\ln i$	1	6
9	$\frac{(i+3)!}{2^i}$	1	8
10	$\frac{(i+1)!}{3^i}$	1	6
11	$\frac{e^{3i}}{(i+1)!}$	1	8
12	$\frac{\sqrt{i+1}}{(i+1)!}$	1	10
13	$\frac{(i+1)!}{2^i}$	1	6
14	$5^i(i+2)!$	1	5

## ЛАБОРАТОРНАЯ РАБОТА № 2 РАБОТА С МАССИВАМИ В VBA

### 2.1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

*Массив* – это упорядоченный набор данных разного типа (числового, текстового и др.). Отдельные элементы массива указываются по их номерам (индексам). Массивы бывают *одномерными* и *двумерными*. С массивами можно выполнять ряд операций: сортировку, поиск минимально (максимально элемента), нахождение суммы, произведения элементов массива и др.

В начале программы массив объявляется следующим образом:

**Dim** *ИмяМассива(Размерность)* **As** *ТипЭлементов*

Например:

**Dim A(9) as Integer** – одномерный массив А из 9 целых чисел;

**Dim B(3,4) As Single** – матрица В из трех строк и четырех столбцов действительных чисел.

При работе с массивами организуют цикл, в котором *НачалоЦикла* – это номер первого элемента, а *КонецЦикла* – номер последнего элемента массива. Шаг по умолчанию равен 1, поэтому его не указывают.

Элементы массива читают и записывают в цикле при помощи оператора **Cells**:

**Cells(Номер\_Строки,Номер\_Столбца)** – **Cells (I,J)**. Здесь **I** – номер строки; **J** – номер столбца. Например:

**Cells(1,2)** соответствует адресу ячейки Range (“B1”);

**Cells(2,3)** – адресу Range (“C2”).

#### **Ввод одномерного массива с листа Excel**

Пусть одномерный массив расположен на листе Excel в диапазоне ячеек, например, A1:A10. Тогда его можно ввести с помощью следующего программного кода:

**For i = 1 To 10**

**M(i)=Cells(i,1)** 'номер строки меняется в цикле, столбец первый (A)

**Next i**

### **Ввод двумерного массива (матрицы) с листа Excel**

Пусть матрица расположена на листе Excel в диапазоне ячеек A1:G5:

```
For i = 1 To 5 'перебираем строки  
  For j = 1 To 7 'перебираем столбцы от A до G  
    M(i,j) = Cells(i,j)  
  Next j  
Next i
```

В случае с матрицей используется так называемый *вложенный цикл*. Работает вложенный цикл следующим образом. Счетчик строк  $i$  получает очередное значение (здесь вначале это 1), затем полностью выполняется цикл со счетчиком  $j$ , т.е. перебираются все элементы  $i$ -той строки. После этого значение счетчика строк увеличивается на единицу ( $i=i+1$ ), и вновь полностью выполняется цикл по  $j$ , перебирающий столбцы матрицы. Процедура завершается после перебора последней строки.

### **Ввод массива с клавиатуры**

Ввод массива фиксированной длины можно также осуществить с клавиатуры. Пусть, например, требуется ввести массив  $M$  из 10 целых чисел:

```
For i = 1 To 10  
  M(i) = InputBox ("Введите элемент массива")  
Next i
```

## **2.2. ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ С МАССИВАМИ**

**Пример 1.** Поиск максимального значения одномерного массива. Элементы массива ввести с клавиатуры.

### **Анализ задачи**

В данной задаче представлен один из примеров обработки введенного массива. Каждый элемент  $A(i)$  массива сравнивается с уже полученным на предыдущих шагах максимумом. В качестве начального значения можно взять любой из элементов  $A(i)$ , но удобнее первый –  $A(1)$ . Циклическую проверку можно начать со второго элемента, сократив число шагов:

```

Dim i As Integer
Dim max As Integer
Dim A(5) As Integer
For i = 1 To 5
    A(i) = InputBox("Введите число - элемент массива")
Next i
max = A(1)
For i = 2 To 5
    If A(i) > max Then
        max = A(i)
    End If
Next i
MsgBox("максимальный элемент массива – " & max)

```

**Пример 2.** Найти сумму, произведение и среднее арифметическое элементов одномерного массива A, состоящего из восьми элементов.

Массив A(8) прочитать с листа Excel из диапазона ячеек **C5:J5**. Сумму записать в ячейку **E7**; среднее арифметическое – в ячейку **K8**. Произведение всех элементов массива вывести в диалоговом окне.

### Решение

1. Оформить лист Excel согласно условию задачи: ввести в диапазон **C5:J5** любые целые числа – как положительные, так и отрицательные (рис. 2.1).

	A	B	C	D	E	F	G	H	I	J	K
1	Пример 1.										
2	<b>Вычислить сумму, произведение и среднее</b>										
3	<b>арифметическое элементов массива.</b>										
4											
5			-5	2	3	5	1	-3	4	1	
6											
7	сумма элементов:				<input type="text"/>						
8	среднее арифметическое элементов массива=									<input type="text"/>	
9											

Рис. 2.1. Оформление листа Excel для решения примера 2

2. Открыть редактор **VBA**. Создать новую процедуру и ввести программный код:

```

Public Sub массив_1()
Dim a(8) As Integer 'массив из 8-ми целых элементов
Dim S, P, i As Integer
Dim srednee As Single
S = 0 'начальная сумма
P = 1 'начальное произведение =1
For i = 1 To 8 'цикл по считыванию элементов массива
    a(i) = Cells(5, i + 2) 'задание эл-тов массива
    S = S + a(i) 'вычисление суммы
    P = P * a(i) 'вычисление произведения
Next
Range("E7") = S 'вывод суммы в ячейку E7
srednee = S / 8 'вычисление среднего арифметического
Range("K8") = srednee 'вывод ср.арифм. в ячейку K8
MsgBox ("произведение= " & P) 'вывод произведения
End Sub

```

**Примечание.** Важно правильно задать адрес первого элемента массива: **Cells(5,i+2)**. В операторе чтения цифра **5** указывает номер строки ячейки, а **i+2** – номер столбца **C** (**C** – это третий столбец, с учетом того, что на первом шаге цикла **i=1**). В цикле изменяется номер столбца (от **C** до **J**).

3. Запустить программу на выполнение и получить результат:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Пример 1.													
2	Вычислить сумму, произведение и среднее													
3	арифметическое элементов массива.													
4														
5			-5	2	3	5	1	-3	4	1				
6														
7	сумма элементов:													
8	среднее арифметическое элементов массива=													
9														
10														
11														
12														
13														

Microsoft Excel

произведение= 1800

OK

Рис. 2.2. Результат решения примера 2

**Пример 3.** Поиск минимального элемента двумерного массива.

#### **Анализ задачи**

В данной задаче каждый элемент массива  $A(i,j)$  сравнивается с уже полученным на предыдущих шагах минимумом. В качестве начального значения минимального элемента можно взять любой из массива  $A(i,j)$ , но удобнее первый -  $A(1,1)$ :

```
Dim i, j As Integer  
Dim min As Integer  
Dim A(4,4) As Integer  
For i = 1 To 4  
  For j = 1 To 4  
    A(i,j) = InputBox("Введите число - элемент массива")  
  Next j  
Next i  
min = A(1,1)  
For i = 1 To 4  
  For j = 1 To 4  
    If A(i,j) < min Then  
      min = A(i,j)  
    End If  
  Next j  
Next i  
Msgbox("минимальный элемент массива – " & min)
```

**Пример 4.** В двумерном массиве **P(6,7)** определить количество положительных элементов и сумму отрицательных элементов.

Массив прочитать с листа Excel из диапазона ячеек **B9:H14**.

Результаты вычислений вывести в диалоговых окнах.

#### **Решение**

1. Оформить лист Excel согласно условию задачи: ввести в диапазон ячеек **B9:H14** любые целые числа (рис. 2.3).

	B	C	D	E	F	G	H
9	-1	-6	7	8	-4	2	-6
10	6	-3	5	4	-3	1	-4
11	13	0	3	0	-2	0	-2
12	20	3	1	-4	-1	-1	0
13	27	6	-1	-8	0	-2	2
14	34	9	-3	-12	1	-3	4

Рис. 2.3. Оформление листа Excel для решения примера 4

2. Открыть редактор VBA. Создать новую процедуру и ввести программный код:

```

Public Sub массив_2()
Dim p(6, 7) As Integer
Dim s, plus, i, j As Integer
plus = 0
s = 0
For i = 1 To 6
For j = 1 To 7
    p(i, j) = Cells(i + 8, j + 1)
    If p(i, j) >= 0 Then
        plus = plus + 1 'подсчет кол-ва положит. элементов
    Else
        s = s + p(i, j) 'вычисление суммы отрицательных эл-тов
    End If
Next
Next
MsgBox ("кол-во положительных эл-тов = " & plus)
MsgBox ("сумма отрицательных эл-тов = " & s)
End Sub

```

**Примечание.** Важно правильно задать адрес первого элемента массива: **P(i,j)=Cells(i+8,j+1)**. В операторе чтения **i+8** – номер строки **9**; **j+1** – буква столбца **B** (второй столбец).

3. Запустить программу на выполнение и получить результат.



### 2.3. ВАРИАНТЫ ЗАДАНИЙ

**Задача 1.** Вычислить сумму, произведение и среднее арифметическое элементов одномерного массива. Определить значение минимального элемента массива. Элементы массива записаны в ячейки Excel.

Сумму и среднее арифметическое элементов записать в ячейки Excel. Произведение и значение минимального элемента массива вывести в диалоговом окне.

*Таблица 2.1*

Вар-т	Диапазон ячеек	Вар-т	Диапазон ячеек
1	B2:B9	8	B3:B11
2	C3:C10	9	C4:C12
3	B2:J2	10	B3:J3
4	C2:K2	11	C3:K3
5	D2:D10	12	D3:D12
6	D2:K2	13	D3:K3
7	E3:E12	14	E2:M2

**Задача 2.** Вычислить количество отрицательных элементов двумерного массива и сумму положительных элементов. Определить значение максимального элемента массива.

Элементы массива записаны в ячейки Excel. Количество отрицательных элементов записать в ячейку Excel. Сумму и значение максимального элемента массива вывести в диалоговом окне.

*Таблица 2.2*

Вар-т	Диапазон ячеек	Вар-т	Диапазон ячеек
1	B2:D5	8	B3:D6
2	C3:E7	9	C4:E8
3	B2:J4	10	B3:J5
4	C2:F5	11	C3:F7
5	D2:F6	12	D3:F7
6	D2:G5	13	D4:G6
7	E2:M8	14	A3:E9

## ЛАБОРАТОРНАЯ РАБОТА № 3 ТАБУЛИРОВАНИЕ ФУНКЦИЙ

### 3.1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Типичным примером циклического процесса является задача табулирования функции, которая сводится к получению таблицы значений функции  $y = f(x)$  некоторой переменной  $x$ , изменяющейся от начального значения  $x_0$  до конечного  $x_k$  с постоянным шагом  $h$ . Данная задача реализуется с помощью цикла с известным количеством повторений, которое определяется по формуле:

$$n = [(x_k - x_0)/h] + 1, \quad (3.1)$$

где  $[a]$  – целая часть от числа  $a$ .

При этом множество пар значений аргумента  $x$  и функции  $f(x)$  вычисляются в теле цикла.

### 3.2. ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ

**Пример 1.** Протабулировать функцию  $y = x^2 e^{-x} \cdot \sin x$  в интервале изменения аргумента  $x$  от -3 до 2 с шагом 0,5.

**1 способ.** Результат вывести в диалоговое окно.

**2 способ.** Результат вывести на лист Excel.

#### Анализ задачи

Введем обозначения:

$a$  - начальное значение переменной  $x$ ;

$b$  - конечное значение  $x$ ;

$h$  - шаг изменения  $x$ .

Тогда в данном примере:  $a = -3$ ;  $b = 2$ ;  $h = 0.5$

#### Способ 1. Решение

1. Оформить лист Excel согласно условию задачи (рис. 3.1).

2. Вызвать редактор VBA, создать новый модуль с именем «Пример\_1»; набрать программный код с применением оператора цикла **For...Next**.

3. Запустить программу на выполнение и получить результат (рис. 3.2).

	A	B	C	D	E
1	Пример 1. Протабулировать функцию				
2					
3	$y = x^2 e^{-x} \cdot \sin x$		в интервале [-3, 2] с шагом 0.5		

Рис. 3.1. Оформление листа Excel для решения примера 1

Программный код для примера 1 (способ 1):

**Dim s As String**

**Dim x, y As Single**

**s = " x y"** *'заголовок таблицы*

**For x = -3 To 2 Step 0.5** *'задание начального и конечного значений аргумента x с указанием шага изменения*

**y = x ^ 2 \* Exp(-x)\*Sin(x)** *'вычисление значения функции  
'формирование строки с результатом вычислений*

**s = s+vbNewLine+Format(x, "0.0") + " " + Format(y, "#.00")**

**Next x**

**MsgBox (s)**

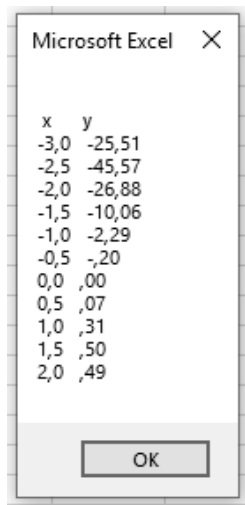


Рис. 3.2. Результат решения примера 1 (способ 1)

## Способ 2. Решение

1. Оформить лист Excel согласно условию задачи:

	A	B	C	D	E	F
1	Пример 1. Протабулировать функцию					$y = x^2 e^{-x} \cdot \sin x$
2						
3		в интервале [-3, 2] с шагом 0.5				
4						
5		x	y			
6						

Рис. 3.3. Оформление листа Excel для примера 1 (способ 2)

2. Вызвать редактор VBA, создать новый модуль с именем «Пример\_1\_2»; набрать программный код:

```
Public Sub Пример_1_2()
```

```
Dim x, y As Single
```

```
Dim i As Integer
```

```
'Начнём вывод значений x и y со строки 6
```

```
i = 6
```

```
For x = -3 To 2 Step 0.5 'начало цикла
```

```
y = x ^ 2 * Exp(-x)*Sin(x)
```

```
'значения аргумента x будем выводить в столбец 2 (B)
```

```
Cells(i,2) = x
```

```
'значения функции y будем выводить в столбец 3 (C)
```

```
Cells(i,3) = y
```

```
i = i + 1 'Изменение номера строки
```

```
Next x 'Изменение значения аргумента x
```

```
End Sub
```

3. Запустить программу на выполнение и получить результат (рис. 3.4).

	A	B	C	D	E	F
1	<b>Пример 1. Протабулировать функцию</b>					$y = x^2 e^{-x} \cdot \sin x$
2						
3		в интервале [-3, 2] с шагом 0.5				
4						
5		<b>x</b>	<b>y</b>			
6		-3	-25,5102			
7		-2,5	-45,568			
8		-2	-26,8754			
9		-1,5	-10,0585			
10		-1	-2,28736			
11		-0,5	-0,19761			
12		0	0			
13		0,5	0,072697			
14		1	0,30956			
15		1,5	0,500785			
16		2	0,49224			

Рис. 3.4. Результат решения примера 1 (способ 2)

**Пример 2.** Протабулировать функцию

$$y = \begin{cases} x^2, & \text{если } x \leq 0 \\ x + 1, & \text{если } x > 0 \end{cases}$$

в интервале изменения аргумента  $x$  от -2 до 2 с шагом 0,4.  
Результат вывести в диалоговое окно.

#### Анализ задачи

1. Введем обозначения:

$a$  - начальное значение переменной  $x$ ;

$b$  - конечное значение  $x$ ;

$h$  - шаг изменения  $x$ .

Тогда в данном примере:  $a = -2$ ;  $b = 2$ ;  $h = 0.4$

2. Для вычисления значений функции в данном примере для каждого значения аргумента  $x$  из заданного интервала необходимо осуществлять проверку принадлежности его одному из двух участков ( $x \leq 0$  или  $x > 0$ ).

Следовательно, внутри цикла необходимо реализовать проверку условия с применением условного оператора **If...Then...Else**.

### Решение

1. Оформить лист Excel согласно условию задачи:

	A	B	C	D	E	F						
1	Пример 2. Протабулировать функцию											
2	$y = \begin{cases} x^2, & \text{если } x \leq 0 \\ x+1, & \text{если } x > 0 \end{cases}$											
3							в интервале [-2, 2] с шагом 0.4					
4												

Рис. 3.5. Оформление листа Excel для решения примера 2

2. Вызвать редактор VBA, создать новый модуль; набрать программный код:

```

Public Sub пример_3()
Dim s As String
Dim x, y As Single
s = " x   y" 'заголовок таблицы
For x = -3 To 2 Step 0.5
y = x ^ 2 * Exp(-x)*Sin(x) 'задание начального и конечного
значений аргумента x с указанием шага изменения
s = s+vbNewLine+Format(x, "0.0") + " " + Format(y, "#.00")
'формирование строки с результатом вычислений
Next x
MsgBox (s)
End Sub

```

3. Запустить программу на выполнение и получить результат (рис. 3.6).

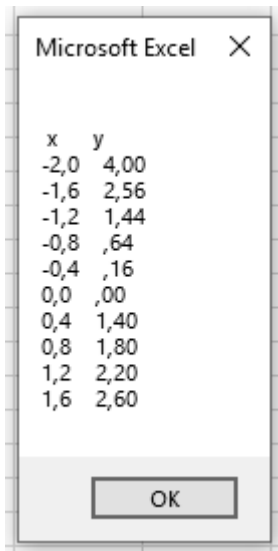


Рис. 3.6. Результат решения примера 2

### 3.3. ВАРИАНТЫ ЗАДАНИЙ

Каждый вариант содержит две задачи.

Задачи решить двумя способами: с выводом результатов в диалоговое окно и на лист Excel.

Предварительно рассчитать по формуле (3.1) количество повторений.

После выполнения заданий необходимо оформить отчет в текстовом редакторе Word с формулировкой заданий, расчетными формулами, программным кодом и результатами вычислений.

**Задача 1.** Протабулировать функции на заданном интервале с заданным шагом (табл. 3.1).

Таблица 3.1

Вариант	Функция	Интервал изменения аргумента	Шаг изменения аргумента
1	$y = x + \frac{4}{x + 0,5}$	[0, 6]	0,5
2	$y = 3 \cdot (x - \sin 2x)$	[-1, 4]	0,5
3	$y = (x + 2) \cdot \sin 3x$	[-2, 3]	0,5
4	$y = \frac{x - \sin 2x}{ x  + 1}$	[-4, 2]	0,5
5	$y = (x + 0,5) \sin 2x$	[-2, 3]	0,5
6	$y = (x - 1) \cdot e^{-x}$	[0, 3]	0,25
7	$y = \sqrt{x} \cdot e^{-x}$	[0, 4]	0,25
8	$y = (x^2 - x) \cdot e^x$	[-3, 2]	0,5
9	$y = \frac{1 - \ln x}{1 + \ln x}$	[1, 6]	0,5
10	$y = \frac{0,5 \cdot x^2 - x + 2}{x^2 + 1}$	[-10, 12]	2
11	$y = \frac{x + 1}{x^2 + 1} e^{-x}$	[-2, 3]	0,5
12	$y = \cos^2 2x - 3 \cdot \sin x$	[-2, 1]	0,25
13	$y = 2^{- x } x$	[-1, 2]	0,25
14	$y = \sqrt[3]{x} \sin x$	[-12, 10]	2
15	$y = \sqrt[3]{x} \cos x$	[-5, 1]	0,5



**Задача 2.** Протабулировать сложную функцию на заданном интервале с заданным шагом (табл. 3.2).

Таблица 3.2

Вар-т	Функция	Интервал	Шаг изменения аргумента
1	$y = \begin{cases} \frac{1+x^2}{\sqrt{1+x^4}}, & \text{если } x \leq 0 \\ \frac{2x + \sin^2 x}{2+x}, & \text{если } x > 0 \end{cases}$	[-10, 12]	2
2	$y = \begin{cases} \sqrt{1+x^2 - \cos^2 x}, & \text{если } x \leq 0 \\ \frac{x}{\sqrt[3]{e^{x+1}}}, & \text{если } x > 0 \end{cases}$	[-2, 3]	0,5
3	$y = \begin{cases} \sqrt{1+x^2}, & \text{если } x \leq 0 \\ \frac{1+x}{1+\sqrt[3]{e^{0.2x}}}, & \text{если } x > 0 \end{cases}$	[-4, 2]	0,5
4	$y = \begin{cases} \frac{3x^2}{1+x^2}, & \text{если } x \leq 0 \\ \sqrt{1 + \frac{2x}{1+x^2}}, & \text{если } x > 0 \end{cases}$	[-3, 2]	0,5
5	$y = \begin{cases} \sqrt{1+ x }, & \text{если } x \leq 0 \\ \frac{1+3x}{2+\sqrt[3]{1+x}}, & \text{если } x > 0 \end{cases}$	[-2, 3]	0,5
6	$y = \begin{cases} \frac{3+\sin x}{1+x^2}, & \text{если } x \leq 0 \\ 2x^2 \cos^2 x, & \text{если } x > 0 \end{cases}$	[-1, 2]	0,25

Продолжение табл. 3.2

7	$y = \begin{cases} \sqrt[3]{1+x^2}, & \text{если } x \leq 0 \\ \frac{1+x}{1+\cos^2 x}, & \text{если } x > 0 \end{cases}$	[-1, 4]	0,5
8	$y = \begin{cases} \frac{1+\cos x}{1+x^2}, & \text{если } x \leq 0 \\ x \cos x, & \text{если } x > 0 \end{cases}$	[-4, 2]	0,5
9	$y = \begin{cases} 3\sin^2 x - \cos x, & \text{если } x \leq 0 \\ \sqrt{2+x^2}, & \text{если } x > 0 \end{cases}$	[-4, 2]	0,5
10	$y = \begin{cases} 3\sin x - \cos^2 x, & \text{если } x \leq 0 \\ 3\sqrt{1+x^2}, & \text{если } x > 0 \end{cases}$	[-12, 10]	2
11	$y = \begin{cases} \sin x - 2\cos x, & \text{если } x \leq 0 \\ \sqrt{1+x^2}, & \text{если } x > 0 \end{cases}$	[-3, 2]	0,5
12	$y = \begin{cases} \frac{ x }{1+x^2} e^{-2x}, & \text{если } x \leq 0 \\ \sqrt{1+x}, & \text{если } x > 0 \end{cases}$	[-2, 4]	0,5
13	$y = \begin{cases} \frac{ x }{1+x^2}, & \text{если } x \leq 0 \\ 2x^2 \cos^2 x, & \text{если } x > 0 \end{cases}$	[-1, 4]	0,5
14	$y = \begin{cases}  x  e^{-2x}, & \text{если } x \leq 0 \\ \frac{1}{\sqrt{1+x^2}}, & \text{если } x > 0 \end{cases}$	[-4, 2]	0,5
15	$y = \begin{cases} \frac{1+\sin x}{1+2\cos x}, & \text{если } x \leq 0 \\ \sqrt{1+x}, & \text{если } x > 0 \end{cases}$	[-2, 3]	0,5

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Агафонов Е.Д.* Прикладное программирование: учебное пособие/ Е.Д. Агафонов, Г.В. Ващенко. – Красноярск: Сибирский федеральный университет, 2015. 112 с.
2. *Воробьева Ф.И.* Информатика. MS Excel 2010: учебное пособие / Ф.И. Воробьева, Е.С. Воробьев. – Казань. КНИТУ, 2014. 100 с.
3. *Грошев А.С.* Информационные технологии: лабораторный практикум/ А.С. Грошев. – Москва, Берлин: Директ-Медиа, 2015. 285 с.
4. *Кузьменко В.Г.* Visual Basic 6. – М.: «Бином-Пресс», 2011. 672 с.
5. Информатика. Visual Basic. Программирование циклических процессов. Графические возможности. Методические указания к лабораторным работам / Национальный металлургический университет «Горный». Сост. Овчинникова Е.Н. – СПб, 2013. 40 с.
6. *Назаров С.В.* Программирование на Visual Basic/ С.В. Назаров, П.П. Мельников. – М.: «Финансы и статистика», 2010. 320 с.
7. *Сергеев А.П.* Microsoft Office 2010. Самоучитель. – М.: Вильямс, 2010. 624 с.
8. *Слепцова Л.Д.* Программирование на VBA в Microsoft Office 2010. – М.: ООО «И.Д. Вильямс», 2010. 432 с.

## СОДЕРЖАНИЕ

Введение.....	3
ЛАБОРАТОРНАЯ РАБОТА № 1.....	4
Решение вычислительных задач с применением операторов цикла ..	4
1.1. Общие сведения.....	4
1.2. Примеры решения вычислительных задач.....	9
1.3. Варианты заданий.....	16
ЛАБОРАТОРНАЯ РАБОТА № 2.....	19
Работа с массивами в VBA .....	19
2.1. Краткие теоретические сведения .....	19
2.2. Примеры решения задач с массивами .....	20
2.3. Варианты заданий.....	25
ЛАБОРАТОРНАЯ РАБОТА № 3.....	26
Табулирование функций.....	26
3.1. Краткие теоретические сведения .....	26
3.2. Примеры решения задач .....	26
3.3. Варианты заданий.....	31
Библиографический список.....	35

**ИНФОРМАТИКА**  
**ОРГАНИЗАЦИЯ ЦИКЛИЧЕСКИХ**  
**ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ В VBA**

*Методические указания к лабораторным работам  
для студентов специальностей 21.05.02, 21.05.04*

Сост.: *Е.Н. Овчинникова, В.Н. Сибирев*

Печатается с оригинал-макета, подготовленного кафедрой  
информатики и компьютерных технологий

Ответственный за выпуск *Е.Н. Овчинникова*

Лицензия ИД № 06517 от 09.01.2002

Подписано к печати 17.03.2020. Формат 60×84/16.  
Усл. печ. л. 2,1. Усл.кр.-отт. 2,1. Уч.-изд.л. 1,8. Тираж 75 экз. Заказ 221. С 43.

Санкт-Петербургский горный университет  
РИЦ Санкт-Петербургского горного университета  
Адрес университета и РИЦ: 199106 Санкт-Петербург, 21-я линия, 2