**Дисциплина: Офисное программирование**

**Группы: 1ПКС-17, 2ПКС-17**

**Дата: 24.09.2020 г.**

**ЗАДАНИЕ:**

1. **Освоить лекционный материал.**
2. **Подчеркнутые определения законспектировать в тетрадь.**
3. **Примеры из лекции запустить в среде разработки VBA.**

**Тема: Циклы. Массивы**

**Циклы**

Решение многих практических задач сводится к выполнению вычислений по одним и тем же зависимостям, но при разных значениях входящих в них величин. Такой вычислительный процесс называется циклическим, а многократно повторяющиеся участки этого процесса называются циклами.

Различают *регулярные* циклы с управляющим параметром (с известным числом повторений), условием окончания которого является достижение параметром цикла своего конечного значения; циклы *итерационные*, в которых условие повторения или окончания цикла задается по некоторому результату, например, пока не будет достигнута точность вычислений. Реализуются циклы с помощью специальных операторов цикла.

***Операторы цикла***

*Оператор цикла* ***While... Wend***

Оператор имеет синтаксис

While *условие*

[*операторы*]

Wend

Синтаксис инструкции While... Wend содержит следующие элементы:

*условие —* обязательный элемент. Числовое выражение или строковое выражение, которое имеет значение True или False.

*операторы—* необязательный элемент. Один или несколько операторов, выполняемых, пока условие имеет значение True.

Этот оператор называют оператором цикла с предусловием. Выполняется оператор While... Wend следующим образом. Если *условие* имеет значение True, выполняются все операторы до инструкции Wend. Затем управление возвращается инструкции While и вновь проверяется *условие.* Если *условие* по-прежнему имеет значение True, процесс повторяется. Если оно не имеет значение True, выполнение возобновляется с инструкции, следующей за инструкцией Wend. В связи с этим элемент *условие* здесь является условием выполнения цикла. Циклы While... Wend могут иметь любую глубину вложенности.

*Оператор цикла* ***Do...Loop***

Оператор Do...Loop также используется для выполнения наборов операторов неопределенное число раз. Оператор имеет синтаксис:

Do

[*операторы*]

Loop Until [*условие*]

Синтаксис инструкции Do Loop содержит следующие элементы:

*условие —* необязательный элемент. Числовое или строковое выражение, которое имеет значение True или False.

*операторы—* один или несколько операторов, выполнение которых повто- ряется, пока условие не приобретет значение True.

Этот оператор называют оператором цикла с постусловием. Выполняется оператор Do...Loop следующим образом. Если *условие* имеет значение False, выполняются все операторы после инструкции Do. Затем управление передается инструкции Until и вновь проверяется *условие.* Если *условие* по-прежнему имеет значение False , процесс повторяется. Если оно имеет значение True, управление передается следующему за инструкцией Loop Until оператору. В связи с этим элемент *условие* здесь является условием выхода из цикла.

*Оператор цикла* ***For...Next***.

Часто при составлении макроса заранее известно количество повторений группы операторов, в таких случаях можно использовать инструкцию For...Next***.***

Оператор For...Next используется для выполнения наборов операторов указанное число раз. Циклы For используют в качестве счетчика переменную, значение которой увеличивается или уменьшается при каждом выполнении цикла на указанное значение.

Синтаксис инструкции For...Next

For *счетчик = начало* То *конец* [step *шаг*]

[*операторы*]

Next [ *счетчик*]

*счетчик —* обязательный элемент. Это должна быть числовая переменная. Она не может иметь тип Boolean или быть элементом массива.

*начало —* обязательный элемент, содержит начальное значение переменной *счетчик.*

*конец —* обязательный элемент, содержит конечное значение переменной *счетчик.*

*шаг —* элемент необязательный, это значение, на которое изменяется счетчик при каждом выполнении тела цикла. Если это значение не задано, по умолчанию *шаг* равен единице. *Шаг* может быть как положительным, так и отрицательным.

*операторы—* необязательный элемент. Один или несколько операторов между For и Next, которые выполняются указанное число раз.

Инструкция For...Next работает следующим образом: начальное значение элемента *счетчик* сравнивается с конечным значением. Если *шаг* положителен и начальное значение меньше конечного или если шаг отрицателен и начальное значение больше конечного, то управление передается внутрь тела цикла. После выполнения всех операторов в теле цикла значение *шаг* добавляется к текущему значению переменной *счетчик.* После этого операторы тела цикла либо выполняются еще раз (на основе того же условия, которое привело кначальному выполнению цикла), либо цикл завершается и выполнение продолжается с оператора, следующей за Next.

Допускается вложение циклов For...Next (один цикл For...Next располагается внутри другого). Счетчик каждого цикла должен иметь уникальное имя.

Пример. Составить таблицу значений функции y=x2 на отрезке [2,8] с шагом 0.5.

Sub Tablica()

Dim x, y As Double, i As Integer

i = 1

Cells(1, 1) = "X": Cells(1, 2) = "Y"

For x = 2 To 8 Step 0.5

y = x ^ 2

i = i + 1

Cells(i, 1) = x: Cells(i, 2) = y

Next x

End Sub

Пример 1. Вычислить сумму квадратов n чисел: 1,4,9,16,25,… . Число n задать произвольно при вводе.

Sub sum()

Dim n, i As Integer, s As Double

n = Val(InputBox("Введите количество слагаемых n"))

s = 0

For i = 1 To n

s = s + i ^ 2

Next i

MsgBox ("Сумма s=" + Str(s#))

End Sub

Пример 2. Вычислить произведение квадратов n чисел: 1,4,9,16,25,… . Число n задать произвольно при вводе.

Sub Proiz()

Dim n, i As Integer, p As Double

n = Val(InputBox("Введите количество слагаемых n"))

p = 1

For i = 1 To n

p = p \* i ^ 2

Next i

MsgBox ("Произведение p=" + Str(p#))

End Sub

Пример 3. Дана числовая последовательность:

Вычислить сумму данной последовательности при n=∝ с точностью 0.000001.

Sub Iter()

Dim a, e, s As Double

a = 1 'первый член ряда

s = a 'сумма ряда

e = Val(InputBox("Введите точность вычислений"))

Do 'начало цикла

a = -a / 2 'вычисляем очередной член последовательности

s = s + a 'накапливаем сумму

Loop Until Abs(a) < e 'конец цикла

MsgBox ("Сумма s=" + Str(s#))

End Sub

***Вложенные циклы***

Внутри одного цикла могут находится один или несколько других циклов. В этом случае охватывающий цикл называется внешним, а вложенные в него циклы – внутренними. Правила организации как внешнего, так и внутренних циклов аналогичны правилам организации простого цикла. Параметры внешнего и внутреннего циклов изменяются не одновременно, т.е. при одном значении параметра внешнего цикла параметр внутреннего последовательно принимает все возможные значения. При организации вложенных циклов необходимо следить за тем, чтобы область действия внутреннего цикла не выходила за область действия внешнего цикла.

Пример 4. Составить таблицу значений функции на отрезке [1,4] c шагом h=0.5.

Sub VlCircle()

Dim x, s, a, b, h As Double, i, n, k As Integer

a = Val(InputBox("Введите а"))

b = Val(InputBox("Введите b"))

h = Val(InputBox("Введите шаг h"))

n = Val(InputBox("Введите количество слагаемых n"))

k = 1

Cells(1, 1) = "X": Cells(1, 2) = "S"

For x = a To b Step h

s = 0

For i = 1 To n

s = s + x / i

Next i

k = k + 1

Cells(k, 1) = x: Cells(k, 2) = s

Next x

End Sub

**Массивы**

Массивом называется упорядоченная последовательность величин, обозначаемая одним именем. Упорядоченность заключается в том, что элементы массива располагаются в последовательных ячейках памяти. Чтобы получить доступ к нужному элементу массива нужно указать имя массива и индекс этого элемента. Имя массива образуется также как имя переменой. Различают одномерные и двумерные массивы. Одномерный массив – это список переменных, двумерный массив – таблица, имеющая строки и столбцы. Элементы одномерного массива снабжаются одним индексом, заключенным в круглые скобки. Он определяет порядковый номер элемента в массиве. Элементы двумерного массива снабжаются двумя индексами, заключенными в квадратные скобки и разделенными запятой. Первый индекс номер строки, второй номер столбца, на пересечении которых расположен элемент в таблице (матрице). Например: A(5)=3, B(2,3)=6. Все используемые массивы должны быть описаны до их использования в программе.

*Оператор описания* ***DIM***

Оператор Dim применяется для описания переменных и выделения для них памяти.

#### Синтаксис инструкции

Dim *имя переменной* [(*индексы*)] [As *тип*]

*имя переменной —* обязательный элемент, содержит имя переменной;

*индексы —* необязательный элемент. Если индексы отсутствуют, то описана простая переменная;

*mun —* необязательный элемент, это тип данных переменной. Если тип данных не указан, по умолчанию переменная получает тип Variant.

Например:

Dim Num As Integer ‘явно объявляет переменную целого типа .

Если необходимо описать массив, то с помощью индексов указывается размерность массива.

Пример:

Dim B(3,3) As Single

Dim A(11) As Integer

Первый оператор объявляет двумерный массив 3\*3 матрицу, состоящую из действительных чисел. Второй оператор объявляет одномерный массив (вектор) из 12 целых чисел, причем по умолчанию первый элемент массива будет А(0) а последний А(11). В этом случае говорят, что 0 - базовый индекс. Можно изменить базовый индекс, написав в области описания модуля инструкцию Option base 1. После этого индексы массивов будут нумероваться с единицы. Другим способом изменения базового индекса является использование ключевого слова To при объявлении массива:

Dim B(1 To 3, 1 To 3) As Single

#### Dim A(1 To 12) As Integer

Массив в программе определяется поэлементно. Например:

#### Dim B(1 To 2, 1 To 3) As Single

B(1,1)=2: B(1,2)=5: B(1,3)=4:

B(2,1)=7: B(2,2)=1: B(2,3)=3

Для определения одномерных массивов можно использовать функцию ***Array***, преобразующую список элементов, разделенных запятыми, в вектор из этих значений, и присваивающую им тип Variant. Например:

#### Dim A As Variant

A= Array(10,40,60)

Здесь A(0)=10, A(1)=40, A(2)=60.

Удобным способом ввода/вывода массивов является использование рабочего листа Excel. Исходный массив считывается из соответствующих ячеек рабочего листа, выводится массив также в ячейки рабочего листа. В этом случае используется оператор цикла ***For…Next*** и свойство Cells() для указания конкретного местоположения элемента массива. Если макрос написан для того же рабочего листа, где располагаются элементы массива, то указание на выбор этого рабочего листа можно опускать.

Рассмотрим некоторые **алгоритмы обработки одномерных массивов**.

*Поиск максимального элемента*

Пример 5. Дан массив A из пяти чисел. Найти максимальный элемент массива и его индекс.

Sub Massiv()

Dim a(5) As Single

Dim max As Single, i, k As Integer

'ввод массива

For i = 1 To 5

a(i) = Cells(1, i + 1)

Next i

max = a(1)

k = 1 'индекс максимального элемента

For i = 1 To 5

If max < a(i) Then max = a(i): k = i

Next i

MsgBox ("Максимальный элемент=" + Str(max!))

MsgBox ("Индекс максимального элемента k=" + Str(k))

End Sub

*Поиск количества элементов по данному правилу*

Пример 6. Дан массив А из пяти чисел. Найти количество положительных элементов массива.

Sub Kol()

Dim a(5) As Single

Dim i, k As Integer

For i = 1 To 5

a(i) = Cells(1, i + 1)

Next i

k = 0

For i = 1 To 5

If a(i) > 0 Then k = k + 1

Next i

MsgBox ("Количество положительных элементов k=" + Str(k))

End Sub

*Нахождение элементов массива по элементам другого массива*

Пример 7. Дан вектор А(5). Найти элементы вектора В(5) по следующему правилу: *bi=sin(ai),I=1,…5.*

Sub NewMassiv()

Dim a(5), b(5) As Single

Dim i As Integer

'ввод массива a(5)

For i = 1 To 5

a(i) = Cells(1, i + 1)

Next i

For i = 1 To 5

b(i) = Sin(a(i))

Next i

'вывод нового массива b(5)

Cells(3, 1) = "Массив b(5)"

For i = 1 To 5

Cells(3, i + 1) = b(i)

Next i

End Sub

Умножение двух векторов

Пример 8. Найти скалярное произведение двух векторов А(5) и С(5).

Sub SkalProiz()

Dim a(5), c(5) As Single

Dim i As Integer, s As Single

'ввод массива a(5)

For i = 1 To 5

a(i) = Cells(1, i + 1).Value

Next i

'ввод массива c(5)

For i = 1 To 5

c(i) = Cells(7, i + 1).Value

Next i

s = 0

For i = 1 To 5

s = s + a(i) \* c(i)

Next i

MsgBox ("Скалярное произведение s=" + Str(s))

End Sub

**Некоторые алгоритмы обработки матриц**

Поиск минимального элемента матрицы

Пример 9. Дана матрица А(2,3). Найти минимальный элемент.

Sub MinMat()

Dim a(2, 3) As Single

Dim min As Single, i, j As Integer

'ввод матрицы

For i = 1 To 2

For j = 1 To 3

a(i, j) = Cells(i + 1, j)

Next j

Next i

min = a(1, 1)

For i = 1 To 2

For j = 1 To 3

If a(i, j) < min Then min = a(i, j)

Next j

Next i

MsgBox ("Минимальный элемент min=" + Str(min))

End Sub

Умножение матрицы на вектор

Пример 10. Дана матрица А(2,3) и вектор В(3). Найти произведение А\*В.

Sub UmnMatNaVec()

Dim a(2, 3) As Single, b(3) As Single, c(2) As Single

Dim s As Single, i, j As Integer

'ввод матрицы

For i = 1 To 2

For j = 1 To 3

a(i, j) = Cells(i + 1, j)

Next j

Next i

'ввод массива

For i = 1 To 3

b(i) = Cells(4, i + 1)

Next i

For i = 1 To 2

s = 0

For j = 1 To 3

s = s + a(i, j) \* b(j)

Next j

c(i) = s

Next i

'вывод нового массива b(5)

Cells(6, 1) = "Массив c(2)"

For i = 1 To 2

Cells(6, i + 1) = c(i)

Next i

End Sub