SF EDUCATION

Программирование на VBA

Одномерные массивы Array

Содержание

1. Что такое массивы
2. Как создавать массивы
3. Динамический массив

1. Что такое массивы

Мы переходим к нашей таблице с учениками и их оценками.

В прошлый раз, когда мы считали отличников, ударников и троечников, мы брали каждую оценку и сравнивали ее с общим значением и после этого выводили результат.

Единственная проблема была в том, что у нас было 250 значений и эти все значения можно было занести в массив.

Массив – это данные, значения которых соответствует определённому значению в массиве, т.е. привязке в массиве.

Можно сказать, что это таблица в таблице. Каждый элемент в массиве пронумерован.

Наш массив будет состоять из количества учеников, и каждый ученик будет иметь свое ФИО и это ФИО будет содержаться в одной ячейке.

Вот как примерно это и выглядит сейчас в столбцах A и B



	A	B
1	№	Фио
2	1	Орлов Артём Андреевич
3	2	Лобанов Даниэль Михайлович
4	3	Волков Дмитрий Андреевич
5	4	Горюнова Ксения Ильинична
6	5	Николаева Вера Олеговна
7	6	Захарова Вероника Григорьевна
8	7	Орлова Александра Романовна
9	8	Дьяконова Анна Ильинична
10	9	Морозов Михаил Платонович
11	10	Смирнов Иван Ильич
12	11	Кириллова Владислава Ярославовна
13	12	Софронова Анастасия Антоновна
14	13	Кравцова Амина Романовна
15	14	Новикова Есения Арсеновна
16	15	Баранова Алиса Кирилловна

Т.е. есть нумерация массива и значения массива. В значениях массива может быть все, что угодно.

2. Как создавать массивы

Создаются массивы, как и обычные переменные с помощью Dim, дальше идет название массива, затем в скобочках прописываем сколько значений внутри этого массива.

Есть три типа задания количества элементов массива, которые мы прописываем в скобочках, т.е. сколько элементов будет в нашем массиве:

1. Просто прописать цифру, на пример 2, это значит, что будет 2 элемента в массиве нулевой и первый;
2. Написание с какого элемента по какой элемент мы задаем. К примеру (1 to 5), т.е. у нас создается массив с пятью элементами внутри;
3. Задать динамический массив, т.е. если мы не знаем сколько внутри массива будет элементов.

Поработаем для начала со вторым типом массива. Нам нужно понять, как записать элементы в этот массив. Для этого мы используем цикл **For i.**

```
Sub massive()

Dim testArray(1 To 5) As Long
For i = 1 To 5
    testArray(i) = Worksheets(2).Range("D" & i + 1)
Next i
End Sub
```

Как только будет первый элемент нашего массива i=1, в первую позицию массива будет записано значение из второго листа нашей книги из диапазона D i+1. Дальше обозначим, что мы “записываем оценки в ячейки массива”. Затем проведем точно такую же работу, только “выводим оценки”.

```
Sub massive()

Dim testArray(1 To 5) As Long
'записываем оценки в ячейки массива
For i = 1 To 5
    testArray(i) = Worksheets(2).Range("D" & i + 1)
Next i
'выводим оценки
For i = 1 To 5
    MsgBox testArray(i)
Next i
End Sub
```

Вот так мы можем записать ячейки в массив и точно также записать в ячейки нашей таблицы значения из массива.

3. Динамический массив

Возможно, будут такие моменты, когда нам нужно будет работать с массивами неизвестной длины.
Для начала научимся как определять размерность массива, для этого нам необходимо знать нижнюю и верхнюю границу массива.

Для определения нижней границы массива используем **LBound.** Для определения верхней границы массива используем **UBound.**

С помощью MsgBox можем выдать такую функцию:

MsgBox LBound (testArray)

MsgBox UBound (testArray)

Как получить размерность массива, т.е. сколько элементов внутри массива, нам достаточно будет получить разность между элементами UBound минус LBound.
MsgBox UBound (testArray) - MsgBox LBound (testArray)

Теперь поработаем с динамическим массивов, заранее скажем, чему равно UBound и LBound.

```
Sub dinmassive()
Dim array2() As Long
Dim низ As Long
Dim вер As Long
Dim i As Integer
низ = 1
вер = 5
ReDim array2(низ To вер)
For i = низ To вер
    MsgBox i
    MsgBox array2(i)
Next i
End Sub
```

Таким образом мы изменили наш массив. Единственное, возникает проблема то, что ReDim удаляет все значения, которые у нас были.

Есть два способа сделать так, чтобы мы изменили размерность массива и сохранили значения:

1. Первый способ – задать еще один массив, т.е. ввести новый массив и после этого часть значений из старого массива перекинуть в новый массив, а старый потом удалить.
2. Второй способ – использовать немного изменённую функцию ReDim, которая будет называть Preserve и тогда все должно отлично измениться.

Как только вы поработали с массивом, а если дальше идет макрос, то нужно закрыть массив и удалить его из памяти. Это делается для ускорения процесса. Для этого достаточно написать Erase и наш массив.

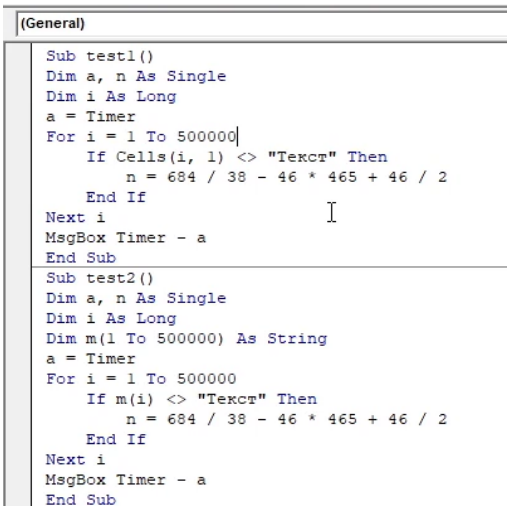
Это позволит:

- Еще раз его использовать, задать его чуть ниже;
- Сэкономить память, потому что значения под массив было выделено.

Почему макросы, которые используют массивы более продуктивные, чем макросы, которые используют данные.

Пример.

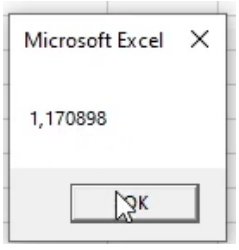
Создадим два Sub.



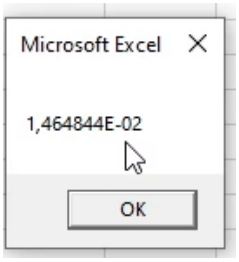
```
(General)
Sub test1()
Dim a, n As Single
Dim i As Long
a = Timer
For i = 1 To 500000
    If Cells(i, 1) <> "Текст" Then
        n = 684 / 38 - 46 * 465 + 46 / 2
    End If
Next i
MsgBox Timer - a
End Sub
Sub test2()
Dim a, n As Single
Dim i As Long
Dim m(1 To 500000) As String
a = Timer
For i = 1 To 500000
    If m(i) <> "Текст" Then
        n = 684 / 38 - 46 * 465 + 46 / 2
    End If
Next i
MsgBox Timer - a
End Sub
```

В первом тесте мы возьмем каждый элемент и будем сравнивать со словом Текст, во втором случае мы возьмем элемент в массиве и его будем сравнивать, посмотрим, насколько это продуктивно на больших данных.

Запускам макрос и видим, что было выполнение на 1,170898 сек, т.е. за это время первый тест у нас выполнил сравнения.



Второй тест сделал сравнение быстрее чем одна сотая секунды.



Т.е. намного эффективней использовать массивы нежели обычные сравнения по элементам. Если данные увеличатся, то пропорционально увеличится используемая мощность.