SF EDUCATION

Программирование на VBA

Вводный урок. Функции и процедуры Часть 1

Содержание

1. Работа с блоком With
2. Добавление страниц
3. Двумерные массивы

1. Работа с блоком With

Работа с блоком With очень важна для быстрого написания некоторых функций, таких, например, как функция «Сортировка массива».
После этого мы научимся создавать функции, которые можно использовать в Excel и передавать другим для использования.

Перейдем в наш файл в Excel.

Чтобы изменить формат одного из столбцов в нашей таблице:

- Мы всегда писали Range, дальше выделяли диапазон, с которым мы хотим работать - “B2 : B251”;
- После этого писали, что обращаемся к этому диапазону через font Name, называем тип шрифта в нашем диапазоне – Arial.
  Range (“B2 : B251”). Font. Name = “Arial”;
- Прописываем в формуле параметры шрифта
  Range (“B2 : B251”). Font. Bold = True;
- Размер шрифта **- Range (“B2 : B251”). Font. Size = 20;**
- Добавим стиль шрифта - **Range (“B2 : B251”). Font. Italic = True;**
- **Range (“B2 : B251”). Font. Underline = True**

И так, мы поменяли параметры надписи в левом столбце нашей таблицы.
Что вернуть все назад, т.е. вернуть шрифту в колонке первоначальные значения, запишем:

Range (“B2 : B251”). Font. Name = “Calibri”

Range (“B2 : B251”). Font. Bold = False

Range (“B2 : B251”). Font. Size = 11

Range (“B2 : B251”). Font. Italic = False

Range (“B2 : B251”). Font. Underline = False

> **Блок With поможет нам обращаться к элементам формулы с одной и той же конструкцией**

Чтобы не переписывать несколько раз одну и ту же конструкцию в однотипных формулах, применим **блок With**:

With Range (“B2 : B251”). Font.

Name = “Calibri”

Bold = False

Size = 11

Underline = False Italic = False

End With

End Sub

2. Добавление страниц

Добавить лист перед текущим листом

В основную формулу добавляем запись:

Sub test ()

Sheets. Add

Worksheets. Add

End sub

Добавить лист после текущего листа

В основную формулу добавляем запись:

Sub test ()

Sheets. Add after : = Sheets (“SH2”)

End sub

Точно также можно написать вместо after – **before:**

```
Sub test ()
Sheets. Add before: = Sheets (“SH2”)
End sub
```

При этом новый лист добавится перед **SH2.**

Добавить лист с определенным именем

В этом случае нам достаточно обратиться к **Add** по имени:

```
Sub test ()
Sheets. Add Name = “test”
End sub
```

Поменять имя рабочего листа

Обратимся к имени второго листа и присвоим ему имя «**test».**

Альтернатива – указываем номер листа и присваиваем имя:

```
Sub test ()
Sheets (2). Add Name = “test 2”
Sheets («Лист 3»). Name = “test 3”
End sub
```

Мы можем и **получить имя листов**, запросив их

```
Sub test ()
MagBox  Sheets (1). Name
End sub
```

Копировать лист

Копировать лист и вставить его перед или после текущего листа:

```
Sub test ()
Sheets (1). Copy after : = Sheets (2)
Sheets (1). Copy before : = Sheets (2)
End sub
```

Как передвинуть листы

Нам достаточно использовать **Move** вместо **Copy**:

```
Sub test ()
Sheets (1). Move after : = Sheets (2)
Sheets (1). Move before : = Sheets (2)
End sub
```

3. Двумерные массивы

Для начала рассмотрим обычный статический двумерный массив.
Напоминаю, что порядковый номер массива начинаются с нуля и заканчивается каким-либо числом – 100, 200… и т.д.

> **Одномерный массив – это по сути своей – строка с множеством элементов. Двумерный массив – к этой строке добавляются столбцы**

Для того, чтобы преобразовать данные в нашей таблице в двумерный массив, для начала изменим нумерацию строк и столбцов.

Далее:

- Зайдем в Visual Basic и зададим массив

  ```
  Sub static_array_2d ()
  Dim a (249, 4) As Integer
  End sub;
  ```

- После запуска макроса внизу листа видим иерархическую таблицу нашего двумерного массива.
  В массиве «А» первая строка нулевая, последняя – 249. Не трудно подсчитать, что в интервале от 0 до 149 имеем 250 чисел – у нас с вами 250 строк;
- В каждой строке имеются столбцы – столбцы в строках.

То есть по сути, двумерный массив – это объединение одномерных массивов в определенную структуру.

Как задаются элементы в этом массиве

Допустим, что у нас есть нулевая строка, и мы хотим добавить туда значения, то нам необходимо написать - **a (0,0).** Это будет элемент, который пересекает нулевую строку и нулевой столбец, должен быть равен 1.

Запишем это формулой:

```
Sub static_array_2d ()
Dim a (249, 4) As Integer
a (0,0) = 11
a (0,1) = 4
a (0,2) = 5
a (0,3) = 5
a (0,4) = 5
End sub;
```

Мы можем поэлементно записать все значения в двумерный массив, чтобы с ними дальше работать. Очевидно, что 250 строк и 6 столбцов прописать вручную очень проблематично.

В этом случае мы можем использовать циклы, а точнее цикл в цикле.

В элемент значения i и j мы будем записывать в ячейки из данной нам структуры.

```
Sub static_array_2d ()
Dim a (249, 4) As Integer
For i = 0 to 249
    For j = 0  to 4
        a (i, j) = Range (Cells (), Cells ())
    Next j
Next i
End sub;
```

При этом для **(Cells (), Cells ())** ставим значения из ячейки:

```
a (i, j) = Range (Cells (3+1, 3+j), Cells (3+1, 3+j))
```

Получаем окончательную формулу:

```
Sub static_array_2d ()
```

```
Dim a (249, 4) As Integer
For i = 0 To 249
    For j = To  to 4
        Range (Cells (3+1, 3+j), Cells (3+1, 3+j)). Select
        a (i, j) = Range (Cells (3+1, 3+j), Cells (3+1, 3+j))
    Next j
Next i
End sub
```

Таким образом, при помощи циклов мы записываем все значения нашей таблицы в двумерный массив.

Теперь при нажатии на любую строку мы можем видеть соответствующие ей значения столбцов.

ИТОГ

Можно записывать в двумерный массив элементы из каких – либо таблиц любой размерности из конкретной ячейки для получения соответствующих значений столбцов.