**Как извлечь уникальные значения из списка**

Это глава из книги Билла Джелена [Гуру Excel расширяют горизонты: делайте невозможное с Microsoft Excel](http://baguzin.ru/wp/?p=12724).

Задача: нужно извлечь все уникальные значения из столбца текстовых данных, которые могут содержать дубли. Подобные процедуры могут быть полезны, например, когда требуется заполнить список или поле со списком уникальных значений для выбора пользователя.

Решение: предположим, что рабочий лист содержит список имен в диапазоне А2:А30 (рис. 1). Ячейка A1 содержит заголовок столбца. Вы можете решить эту проблему вручную, пройдя по меню: *ДАННЫЕ* –> *Сортировка и фильтр* –> *Дополнительно*. Откроется диалоговое окно Расширенный фильтр (рис. 2). Выберите опцию *Скопировать результат в другое место*, задайте исходный диапазон $A$1:$A$30, *Диапазон условий* оставьте пустым, в поле *Поместить результат в диапазон* установите ссылку $H$1, выберите *Только уникальные записи,* нажмите Оk.



Рис. 1. Диалоговое окно *Расширенный фильтр*

Список уникальных имен (вместе с заголовком) появится в столбце Н (рис. 2). Если требуется, вы можете отсортировать список в алфавитном порядке.



Рис. 2. Список уникальных имен

Альтернативное решение: чтобы извлечь уникальные значения из списка можно применить код VBA. Рассмотрите два подхода: один с использованием объекта Collection (коллекция), другой – помощью объекта Dictionary (словарь). Эти подходы основаны на одном и том же механизме, использующем тот факт, что и коллекция, и словарь не могут содержать дубли.

**1) Код VBA на основе объекта Collection.**

Sub GetUnique\_Collection()

 Dim SourceRng As Range

 Dim UniqColl As New Collection

 Set SourceRng = Range("A2:A30")

 On Error Resume Next

 For Each cell In SourceRng.Cells

 UniqColl.Add cell.Value, cell.Value

 Next

 On Error GoTo 0

 'Sort the collection (optional)

 ReDim UniqArray(1 To UniqColl.Count)

 For i = 1 To UniqColl.Count

 UniqArray(i) = UniqColl(i)

 Next

 Range("H1").Resize(UniqColl.Count, 1).Value = WorksheetFunction.Transpose(UniqArray)

 UserForm1.ComboBox1.List = UniqArray

 'UserForm1.Show

End Sub

Этот код создает новую коллекцию – UniqColl, и перебирает все значения в списке имен, пытаясь внести каждое имя в коллекцию. Обратите внимание, что в строке кода UniqColl.Add cell.Value, cell.Value содержатся две ссылки на значение ячейки – cell.Value. Это происходит потому, что первые два аргумента для метода Add – это Value и Key. Коллекция не может содержать дубли.

Во время выполнения первого цикла For...Next в случае ошибки программа перенаправляет выполнение кода снова в начало цикла. Так что всякий раз, когда код обнаруживает повторяющееся значение (которое уже имеется в коллекции, и которое не может быть добавлен к ней), программа просто переходит к следующему элементу, вместо того чтобы остановиться и сообщить об ошибке.

Когда все элементы добавлены в коллекцию, код создает массив UniqArray такого же размера, как и коллекция UniqColl. Массив необходим для передачи содержимого в таблицу. Обратите внимание на использование функция Transpose при передаче массива в столбец листа Excel из кода VBA; это необходимо, потому что UniqArray представляет собой горизонтальный массив.

Перед передачей уникального списка UniqArray на лист Excel, данные желательно отсортировать в алфавитном порядке. Для этого просто вставьте следующий код после комментария 'Sort the collection (optional):

 For i = 1 To UniqColl.Count - 1

 For j = i + 1 To UniqColl.Count

 If UniqColl(i) > UniqColl(j) Then

 Temp1 = UniqColl(i)

 Temp2 = UniqColl(j)

 UniqColl.Add Temp1, before:=j

 UniqColl.Add Temp2, before:=i

 UniqColl.Remove i + 1

 UniqColl.Remove j + 1

 End If

 Next j

 Next i

**2) Код VBA на основе объекта Dictionary.**

Sub GetUnique\_Dictionary()

Dim UniqueDic As Object

Dim cell As Range

Set UniqueDic = CreateObject("Scripting.Dictionary")

For Each cell In Range("A2:A30")

 If Not UniqueDic.Exists(cell.Value) Then

 UniqueDic.Add cell.Value, cell.Value

 End If

Next

UniqArray = UniqueDic.Items

Range("H1").Resize(UniqueDic.Count, 1).Value = WorksheetFunction.Transpose(UniqArray)

UserForm1.ComboBox1.List = UniqArray

'UserForm1.Show

End Sub

Этот код работает аналогично предыдущей программе за исключением следующих особенностей:

* Вместо перехвата ошибок (в случае дублей), используется свойство Exists объекта Dictionary, которое решает, было ли значение ранее добавлено в словарь и должно ли оно быть пропущено.
* Содержание словаря переводится в UniqArray одной командой: UniqArray = UniqueDic.Items

Вместо передачи уникальных элементов списка на лист Excel, вы можете заполнить пользовательскую форму *Поле со списком* (combo box) используя инструкцию: UserForm1.ComboBox1.List=UniqArray.[[1]](#footnote-1)

Резюме: вы можете извлечь уникальные значения из списка, содержащего дубли. Данные могут быть помещены на лист Excel или в пользовательскую форму Поле со списком.

Источники: [здесь](http://www.mrexcel.com/forum/excel-questions/8485-load-only-one-instance-no-duplicates.html), [здесь](http://www.mrexcel.com/forum/excel-questions/217977-remove-duplicates-combo-box.html) и [здесь](http://www.mrexcel.com/forum/excel-questions/41643-find-unique-values-column.html).

1. У меня пользовательская форма не заработала – *Прим. Багузина* [↑](#footnote-ref-1)