

## Функции рабочих листов Excel при программировании на VBA

Настоящая заметка продолжает знакомство с VBA. В ней представлены некоторые пользовательские функции, которые можно применять в формулах рабочего листа.<sup>1</sup> Помните, что эти процедуры функций необходимо определить в модуле VBA, а не в модуле кода соответствующей рабочей книги, листа или пользовательской формы.

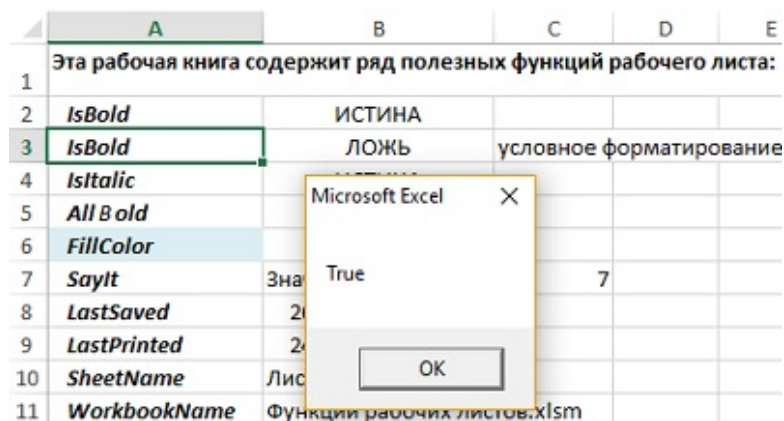


Рис. 1. Активная ячейка имеет полужирное начертание, выполненное с помощью условного форматирования

### Получение информации о форматировании ячейки

Можно написать пользовательскую функцию, возвращающую информацию о форматировании ячейки. Такие функции используются при сортировке данных на основе форматирования (например, в случае, когда ячейки, выделенные полужирным шрифтом, должны располагаться рядом).

Предупреждение. Эти функции не всегда обновляются автоматически — изменение форматирования не приводит к пересчету формул Excel. Чтобы вызвать глобальный пересчет формул (и обновить все пользовательские функции), нажмите клавиши <Ctrl+Alt+F9>. В функцию можно также добавить следующий оператор: `Application.Volatile`. При наличии этого оператора пересчет функции производится после нажатия клавиши <F9>.

Следующая функция возвращает ИСТИНА, если аргумент, состоящий из одной ячейки, выделен полужирным шрифтом. Если диапазон передается в качестве аргумента, функция использует его верхнюю левую ячейку.

```
Function IsBold(cell) As Boolean
' Возвращает TRUE, если ячейка выделена полужирным стилем
  IsBold = cell.Range("A1").Font.Bold
End Function
```

Эта функция работает только с явно заданным форматированием (не может применяться с условным форматированием). В Excel 2010 появился новый объект – `DisplayFormat`. Он учитывает наличие условного форматирования. Функция `IsBold2` может обрабатывать полужирный формат, являющийся результатом условного форматирования. Свойство `DisplayFormat` возвращает ошибку #ЗНАЧ! при использовании в пользовательских функциях. Однако свойство работает в процедурах, вызываемых из VBA. Если выбрать в качестве активной ячейку A3, отформатированную с помощью условного форматирования, и вызвать процедуру `Cl`, появится окно, подтверждающее, что активная ячейка имеет полужирное начертание (рис. 1).

```
Function IsBold2() As Boolean
' Возвращает TRUE, если ячейка выделена полужирным стилем
' даже условным форматированием
  IsBold2 = ActiveCell.DisplayFormat.Font.Bold
End Function
```

<sup>1</sup> По материалам книги [Джон Уокенбах. Excel 2010. Профессиональное программирование на VBA](#). – М: Диалектика, 2013. – С. 362–375.

Следующая функция возвращает ИСТИНА, если используемая в качестве аргумента ячейка выделена курсивом.

```
Function IsItalic(cell) As Boolean
'   Возвращает TRUE, если ячейка выделена курсивом
    IsItalic = cell.Range("A1").Font.Italic
End Function
```

Обе предыдущие функции возвращают ошибку, если ячейка имеет смешанное форматирование. Функция, приведенная ниже, возвращает ИСТИНА только тогда, когда все символы в ячейке выделены полужирным шрифтом.

```
Function AllBold(cell) As Boolean
'   Возвращает TRUE, если все символы в ячейке выделены полужирным
    AllBold = Not IsNull(cell.Font.Bold)
End Function
```

Следующая функция возвращает целое число, соответствующее индексу цвета заливки ячейки. Если ячейка не имеет заливки, то функция возвращает значение 4142. Эта функция не может использоваться для определения цветов заливки таблиц (которые создаются с помощью команды Ctrl+T) или сводных таблиц. В подобных случаях воспользуйтесь объектом DisplayFormat.

```
Function FillColor(cell) As Integer
'   Возвращает целое число, соответствующее фону ячейки
    Application.Volatile
    FillColor = cell.Range("A1").Interior.ColorIndex
End Function
```

### Беседа с рабочим листом

Функция SayIt применяет синтезатор речи Excel для озвучивания аргумента.

```
Function SayIt(txt)
    Application.Speech.Speak (txt)
    SayIt = txt
End Function
```

Эта функция носит развлекательный характер, но может использоваться и в серьезных целях. Например, ее можно включить в следующую формулу:

=ЕСЛИ(СУММ(A:A)>25000;SayIt("Цель достигнута "))

Если сумма значений в столбце A превышает 25 000, вы услышите синтезированный голос, сообщающий о том, что цель достигнута. Метод Speak можно также включить в конец длинной процедуры, и компьютер известит вас о том, что выполнение процедуры завершено.

### Отображение даты сохранения файла или вывода файла на печать

Рабочая книга Excel содержит несколько встроенных свойств документа, к которым можно получить доступ с помощью свойства BuiltinDocumentProperties объекта Workbook. Следующая функция возвращает дату и время последнего сохранения рабочей книги.

```
Function LastSaved()
    Application.Volatile
    LastSaved = ThisWorkbook. _
        BuiltinDocumentProperties("Last save time")
End Function
```

К свойствам можно обратиться по номеру, например, BuiltinDocumentProperties(5), или по имени, как в примере выше. Получить перечень свойств можно с помощью процедуры BDPNames(), код которой приведен в приложенном Excel-файле.

Значения, возвращаемые этой функцией, совпадают со значениями даты и времени, которые отображаются в разделе *Связанные даты* после выбора команды *Файл* → *Сведения* (рис. 2). Обратите внимание, что на значения даты и времени оказывает влияние свойство *AutoSave*. Поэтому *время последнего сохранения* необязательно имеет отношение ко времени сохранения файла пользователем.

**Сведения**

Функции рабочих листов

D: » D:\орбох » !Сайт » 6\_Эффективность » Уокенбах\_VBA » 11.5. Функции Excel

**Защита книги**  
Контролируйте типы изменений, которые пользователи могут вносить в эту книгу.

**Проверка книги**  
Перед публикацией этого файла не забудьте, что он содержит:

- Свойства документа, путь к принтеру, имя автора, связанные даты и абсолютный путь
- Настраиваемые XML-данные
- Макросы, формы и элементы ActiveX
- Содержимое, при чтении которого люди с ограниченными возможностями будут испытывать затруднения

**Версии**

- Сегодня, 11:14 (автосохранение)
- Сегодня, 11:03 (автосохранение)
- Сегодня, 10:51 (автосохранение)
- Сегодня, 10:38 (автосохранение)
- Сегодня, 10:26 (автосохранение)

**Параметры просмотра в браузере**  
Укажите, что именно смогут видеть пользователи при просмотре этой книги в Интернете.

**Свойства**

Размер: 37,2Кбайт  
Название: Функции рабочих листов  
Теги:  
Категории: Excel 2010 Power Programmi...

**Связанные даты**

Изменено	Сегодня, 11:14
Создано	26.01.1999 1:05
Напечатано	24.08.2006 22:27

**Связанные пользователи**

Автор: Сергей Багузин  
Кем изменено: Сергей Багузин

**Связанные документы**

Открыть расположение файла  
Показать все свойства

Рис. 2. Сведения о рабочей книге

Следующая функция возвращает дату и время последнего вывода рабочей книги на печать или предварительного просмотра рабочей книги. Если рабочая книга никогда не печаталась и не просматривалась, функция возвращает ошибку #ЗНАЧ. При использовании двух последних функций, чтобы получить текущие значения данных свойств (чтобы обновить значения в ячейках), нажмите F9.

```
Function LastPrinted()  
    Application.Volatile  
    LastPrinted = ThisWorkbook.BuiltinDocumentProperties(10)  
End Function
```

Не ко всем свойствам можно получить доступ с помощью `BuiltinDocumentProperties`. Например, при попытке получить доступ к свойству `Number of Bytes`, указывающему размер файла, будет генерироваться ошибка.

### Основы иерархии объектов

Объектная модель Excel представляет собой определенную структуру: одни объекты содержатся в других объектах. На вершине этой иерархии находится объект `Application`. Excel содержит другие объекты, в которые, в свою очередь, вложены более низкоуровневые объекты и т.д. Следующая иерархия показывает, как в этой структуре представлен объект `Range`.

```
Объект Application  
    Объект Workbook  
        Объект Worksheet  
            Объект Range
```

Следующая функция получает один аргумент (диапазон) и возвращает имя рабочего листа, который содержит указанный диапазон. При этом используется свойство `Parent` объекта `Range`. Свойство `Parent` возвращает объект, содержащий объект `Range`.

```
Function SheetName(ref) As String  
    SheetName = ref.Parent.Name
```

```
End Function
```

Следующая функция возвращает название рабочей книги для конкретной ячейки. Обратите внимание, что эта функция использует свойство Parent дважды. Первое свойство Parent возвращает объект Worksheet, а второе свойство Parent возвращает объект Workbook.

```
Function WorkbookName(ref) As String
    WorkbookName = ref.Parent.Parent.Name
End Function
```

Следующая функция переносит это упражнение на следующий логический уровень, обращаясь к свойству Parent трижды. Такая функция возвращает имя объекта Application для заданной ячейки. Указанная функция всегда будет возвращать значение Microsoft Excel.

```
Function AppName(ref) As String
    AppName = ref.Parent.Parent.Parent.Name
End Function
```

### Подсчет количества ячеек между двумя значениями

Следующая функция возвращает количество значений в диапазоне (первый аргумент), которые попадают в область, заданную вторым и третьим аргументами.

```
Function CountBetween(InRange, num1, num2) As Long
'    Подсчитывает количество значений между num1 и num2
    With Application.WorksheetFunction
        If num1 <= num2 Then
            CountBetween = .CountIfs(InRange, ">=" & num1, _
                InRange, "<=" & num2)
        Else
            CountBetween = .CountIfs(InRange, ">=" & num2, _
                InRange, "<=" & num1)
        End If
    End With
End Function
```

Обратите внимание, что эта функция вызывает функцию Excel СЧЁТЕСЛИ (COUNTIFS). По сути, функция CountBetween является «оболочкой», которая может упростить формулы. (Но, наверное, не в этом учебном примере, для которого вполне достаточно стандартной функции Excel =СЧЁТЕСЛИМН(A17:A27;"<=5";A17:A27;">=2") Поскольку функция СЧЁТЕСЛИ появилась в Excel 2007, приведенный код не будет работать с более ранними версиями Excel.

### Определение последней непустой ячейки в столбце или в строке

Функция LastInColumn возвращает содержимое последней непустой ячейки в столбце, а LastInRow – содержимое последней непустой ячейки в строке. В качестве единственного аргумента эти функции используют диапазон. Причем в качестве диапазона может использоваться, как весь столбец, так и вся строка. Если же в качестве аргумента не используется вся строка либо весь столбец, задействуется строка или столбец, в котором находится верхняя левая ячейка диапазона. Например, следующая формула возвращает последнее значение в столбце B: =LastInColumn(B5). Следующая формула возвращает последнее значение в строке 7: =LastInRow (C7 :D9).

```
Function LastInColumn(rng As Range)
'    Возвращает содержимое последней ячейки в столбце
    Dim LastCell As Range
    Application.Volatile
    With rng.Parent
        With .Cells(.Rows.Count, rng.Column)
            If Not IsEmpty(.Value) Then
                LastInColumn = .Value
            ElseIf IsEmpty(.End(xlUp)) Then
                LastInColumn = ""
            Else
```

```

        LastInColumn = .End(xlUp).Value
    End If
End With
End With
End Function

```

Оператор `Application.Volatile` вызывает выполнение функции всякий раз, когда пересчитываются формулы на рабочем листе. Оператор `Rows.Count` возвращает количество строк на рабочем листе (используется именно он, а не жестко заданное значение, из соображений совместимости; новые версии Excel могут включать большее количество строк на рабочем листе). Ссылка `rng.Column` возвращает номер столбца левой верхней ячейки в аргументе `rng`. Благодаря ссылке `rng.Parent` функция работает корректно, даже если аргумент `rng` ссылается на другой лист или рабочую книгу. Метод `End` (с аргументом `xlUp`) эквивалентен переходу к последней ячейке столбца и нажатию <End> и <T>. Функция `IsEmpty` проверяет, пуста ли ячейка. Если ячейка пуста, функция возвращает пустую строку. Без этого оператора пустой ячейке соответствовал бы результат 0.

Код функции `LastInRow` можно найти в модуле VBA приложенного Excel-файла.

### Соответствует ли строка шаблону

Следующая функция возвращает значение ИСТИНА, если строка соответствует заданному шаблону. Функция представляет собой «оболочку», позволяющую использовать в формулах оператор VBA `Like`.

```

Function IsLike(text As String, pattern As String) As Boolean
'   Возвращает true, если первый аргумент равен второму
    IsLike = text Like pattern
End Function

```

Функция `IsLike` принимает два аргумента: `text` — текстовая строка или ссылка на ячейку, содержащую текстовую строку; `pattern` — строка, содержащая групповые символы согласно таблице (рис. 3).

Символ(ы) в шаблоне	Соответствует в тексте
?	Любой отдельный символ
*	Ноль и больше символов
#	Любая отдельная цифра (0–9)
[список_символов]	Любой отдельный символ из список_символов
[!список_символов]	Любой отдельный символ, не принадлежащий список_символов

Рис. 3. Групповые символы

Представленная ниже формула возвращает ИСТИНА, так как \* соответствует любому количеству символов. Она возвращает ИСТИНА, если первый аргумент — любой текст, начинающийся с g.

```
=IsLike("guitar", "g*")
```

Следующая формула возвращает ИСТИНА, так как ? соответствует любому отдельному символу. Если бы первым аргументом функции был "Unit12", то функция возвращала бы ЛОЖЬ.

```
=IsLike("Unit1", "Unit?")
```

Следующая формула возвращает ИСТИНА, так как первый аргумент является одним из символов списка во втором аргументе.

```
=IsLike("a", "[aeiou]")
```

Следующая формула возвращает ИСТИНА, если ячейка A1 содержит один из символов: a, e, i, o, u, A, E, I, O, U. При использовании функции ПРОПИСН (UPPER) в аргументе функция становится нечувствительной к регистру.

```
=IsLike(ПРОПИСН(A1), ПРОПИСН("[aeiou]"))
```

Следующая формула возвращает ИСТИНА, если в ячейке A1 находится значение, начинающееся с 1 и состоящее ровно из трех цифр (т.е. любое целое число от 100 до 199).



=IsLike(A1, "1##")

## Возвращение из строки n-го элемента

ExtractElement — специальная функция рабочего листа, которая извлекает элемент из текстовой строки.

```
Function ExtractElement(txt, n, Separator) As String
' Возвращает n-й элемент в строке, причем элементы
' разделены выбранным символом-разделителем
Dim AllElements As Variant
AllElements = Split(txt, Separator)
ExtractElement = AllElements(n - 1)
End Function
```

В этой процедуре используется VBA-функция Split, возвращающая массив констант, из которого состоит текстовая строка. Массив начинается с нулевого элемента (а не с первого), поэтому текущий элемент имеет индекс  $n - 1$ .

Например, если ячейка содержит следующий текст "123-456-789-0133-8844", вы можете использовать функцию ExtractElement для извлечения любых подстрок между дефисами. Следующая формула возвращает 0133, т.е. четвертый элемент в строке.

=ExtractElement("123-456-789-0133-8844",4,"-")

Функция ExtractElement принимает три аргумента:

- txt — текстовая строка, из которой извлекается подстрока (символьная строка или ссылка на ячейку);
- n — целое число, представляющее номер извлекаемого элемента;
- Separator — отдельный символ, используемый как разделитель.

Если в качестве символа-разделителя задать пробел, то несколько пробелов подряд будут рассматриваться как один, что не всегда соответствует требованиям. Если  $n$  превышает количество элементов в строке, функция возвращает пустую строку.

## Преобразование чисел в текст

Функция SpellDollars возвращает текст, в который преобразуется исходное число — сумма в долларах и центах. Например, формула =SpellDollars(23,45) возвращает строку "двадцать три и 45/100 доллара" (рис. 4). Обратите внимание, что отрицательные числа заключаются в круглые скобки. Код функции SpellDollars приведен в приложенном Excel-файле.

	А	В
1	23,45	Двадцать-Три и 45/100 долларов
2	32	Тридцать-Два и 00/100 долларов
3	37,56	Тридцать-Семь и 56/100 долларов
4	-32	(Тридцать-Два и 00/100 долларов)
5	-26,44	(Двадцать-Шесть и 44/100 долларов)
6	1,8754	Один и 88/100 долларов
7	1000000	Один Миллион и 00/100 долларов
8	1000000000	Один Миллиард и 00/100 долларов

Рис. 4. Примеры использования функции SpellDollars

## Универсальная функция

Можно сделать так, чтобы одна функция рабочего листа работала как несколько функций. StatFunction имеет два аргумента: диапазон (rng) и операция (op). В зависимости от значения аргумента op функция возвращает значение, вычисленное с помощью одной из следующих функций Excel: СУММ, СРЗНАЧ, МЕДИАНА, МОДА, СЧЁТ, МАКС, МИН, ДИСП, СТАНДОТКЛОН. Например, результат формулы =StatFunction(\$A\$33:\$A\$37;C36) зависит от содержимого ячейки C36, в которой должна быть текстовая строка с именем одной из допустимых функций. В нашем случае — СЧЁТ. Код функции StatFunction приведен в модуле VBA приложенного Excel-файла.

	A	B	C	D
32	<b>StatFunction</b>		26	СУММ
33		5	5,2	СРЗНАЧ
34		4	4	МЕДИАНА
35		11	3	МОДА
36		3	5	СЧЁТ
37		3	11	МАКС
38			3	МИН
39			11,2	ДИСП
40			3,346640106	СТАНДОТКЛОН

Рис. 5. Примеры использования функции StatFunction

### Функция SheetOffset

В Excel ограничена поддержка «трехмерных рабочих книг». Например, чтобы сослаться на другой рабочий лист в книге, включите в формулу имя рабочего листа. Данная проблема будет оставаться незначительной до тех пор, пока вы не попытаетесь скопировать формулу из одного листа в другой. Скопированные формулы продолжают ссылаться на первоначальное имя рабочего листа, и ссылки на листы не изменяются, как это происходит в реальной трехмерной рабочей книге.

Функция VBA `SheetOffset` обеспечивает установку относительных ссылок на рабочие листы. Например (рис. 6), можно сослаться на ячейку A1 следующего рабочего листа с помощью такой формулы: `=SheetOffset(1; A1)`. Первый аргумент представляет лист и может быть положительным, отрицательным или нулевым. Второй аргумент должен быть ссылкой на одну ячейку. Можете скопировать эту формулу в другие листы, и в скопированных формулах будет использована относительная ссылка.

=SheetOffset(1;A1)/2				
	A	B	C	D
1	123			
2				
	Лист1	Рис. 4	Лист2	Лист3

	A	B	C	D
1	246			
2				
	Лист1	Рис. 4	Лист2	Лист3

Рис. 6. Пример использования функции SheetOffset

```
Function SheetOffset(Offset As Long, Optional cell As Variant)
' Возвращает содержимое ячейки в виде ссылки, в смещении листа
Dim WksIndex As Long, WksNum As Long
Dim wks As Worksheet
Application.Volatile
If IsMissing(cell) Then Set cell = Application.Caller
WksNum = 1
For Each wks In Application.Caller.Parent.Parent.Worksheets
If Application.Caller.Parent.Name = wks.Name Then
SheetOffset = Worksheets(WksNum + _
Offset).Range(cell(1).Address)
Exit Function
Else
WksNum = WksNum + 1

```

```

    End If
  Next wks
End Function

```

### Возвращение максимального значения всех рабочих листов

Если необходимо определить максимальное значение в ячейке B1 в нескольких рабочих листах, используется следующая формула: =МАКС(Лист1:Лист4!B1). Эта формула возвращает максимальное значение ячейки B1 для листов Лист1, Лист4 и всех листов между ними.

Что же произойдет, если добавить после листа Лист4 новый лист – Лист5? Формула не будет автоматически изменена, поэтому ее необходимо отредактировать, чтобы включить ссылку на новый лист: =МАКС(Лист1:Лист5!B1).

Функция VBA MaxAllSheets получает аргумент (одна ячейка) и возвращает максимальное значение в этой ячейке во всех рабочих листах данной книги. При добавлении нового листа редактировать формулу не нужно.

```

Function MaxAllSheets(cell)
  Dim MaxVal As Double
  Dim Addr As String
  Dim Wksht As Object
  Application.Volatile
  Addr = cell.Range("A1").Address
  MaxVal = -9.9E+307
  For Each Wksht In cell.Parent.Parent.Worksheets
    If Wksht.Name = cell.Parent.Name And _
      Addr = Application.Caller.Address Then
      ' Исключение циклической ссылки
    Else
      If IsNumeric(Wksht.Range(Addr)) Then
        If Wksht.Range(Addr) > MaxVal Then _
          MaxVal = Wksht.Range(Addr).Value
        End If
      End If
    End If
  Next Wksht
  If MaxVal = -9.9E+307 Then MaxVal = 0
  MaxAllSheets = MaxVal
End Function

```

Оператор For Each использует для доступа к рабочей книге выражение:

cell.Parent.Parent.Worksheets. «Родителем» ячейки является рабочий лист, «родителем» рабочего листа — рабочая книга. Следовательно, цикл For Each проходит по всем рабочим листам в книге. Первый оператор If внутри цикла проверяет, содержит ли ячейка, которая проверяется в данный момент, функцию. Если содержит, то ячейка игнорируется во избежание циклической ссылки.

Функцию можно изменить, чтобы она определяла минимальное, среднее значение, или сумму.

### Возвращение массива случайных целых чисел без повторов

Функция RandomIntegers возвращает массив натуральных чисел без повторов. Она предназначена для применения в формуле массива в нескольких ячейках: {=RandomIntegers ()}. Формула вводится в диапазон с помощью комбинации клавиш <Ctrl+Shift+Enter>. Она возвращает массив натуральных чисел без повторов, упорядоченных произвольным образом. Если диапазон содержит 35 ячеек, числа принимают значения от 1 до 35 (рис. 7).



	A	B	C	D
1	17			
2	11			
3	13			
4	22			
5	31			
6	14			
7	5			
8	16			
9	27			
10	23			
11	12			
12	15			
13	26			
14	4			
15	33			
16	25			
17	7			
18	34			
19	35			
20	3			
21	30			
22	18			
23	20			
24	9			
25	2			
26	19			
27	24			
28	28			
29	8			
30	32			
31	6			
32	21			
33	10			
34	29			
35	1			
36				

Рис. 7. Пример использования функции RandomIntegers

```

Function RandomIntegers()
    Dim FuncRange As Range
    Dim V() As Variant, ValArray() As Variant
    Dim CellCount As Double
    Dim i As Integer, j As Integer
    Dim r As Integer, c As Integer
    Dim Temp1 As Variant, Temp2 As Variant
    Dim RCount As Integer, CCount As Integer
    ' Создает объект Range
    Set FuncRange = Application.Caller
    ' Возвращает ошибку, если значение FuncRange слишком велико
    CellCount = FuncRange.Count
    If CellCount > 1000 Then
        RandomIntegers = CVErr(xlErrNA)
        Exit Function
    End If
    ' Присваивание значений переменным

```

```

RCount = FuncRange.Rows.Count
CCount = FuncRange.Columns.Count
ReDim V(1 To RCount, 1 To CCount)
ReDim ValArray(1 To 2, 1 To CellCount)
' Заполнение массива случайными числами
' и последовательными целыми числами
For i = 1 To CellCount
    ValArray(1, i) = Rnd
    ValArray(2, i) = i
Next i
' Сортировка ValArray по случайным числовым размерам
For i = 1 To CellCount
    For j = i + 1 To CellCount
        If ValArray(1, i) > ValArray(1, j) Then
            Temp1 = ValArray(1, j)
            Temp2 = ValArray(2, j)
            ValArray(1, j) = ValArray(1, i)
            ValArray(2, j) = ValArray(2, i)
            ValArray(1, i) = Temp1
            ValArray(2, i) = Temp2
        End If
    Next j
Next i
' Помещение рандомизованных значений в массив V
i = 0
For r = 1 To RCount
    For c = 1 To CCount
        i = i + 1
        V(r, c) = ValArray(2, i)
    Next c
Next r
RandomIntegers = V
End Function

```

### Расположение значений диапазона в произвольном порядке

Функция RangeRandomize получает в качестве аргумента диапазон и возвращает массив, содержащий этот диапазон с произвольно переставленными значениями (рис. 8). Код функции приведен во вложенном Excel-файле. В диапазон B2:B11 введена формула массива:

`{=RangeRandomize (A2:A11)}`

Эта формула возвращает содержимое диапазона A2:A11, но в случайном порядке.

	А	В	С
1	Алфавитный порядок	Случайный порядок	
2	Жираф	Игуана	
3	Игуана	Жираф	
4	Кенгуру	Лемур	
5	Кошка	Слон	
6	Лев	Лисица	
7	Лемур	Кошка	
8	Лисица	Лев	
9	Павиан	Кенгуру	
10	Попугай	Павиан	
11	Слон	Попугай	
12	Собака	Собака	
13			

Рис. 8. Функция RangeRandomize возвращает содержимое диапазона в случайном порядке