

Операции с книгами и листами

Вам приходилось когда-нибудь работать с книгой Excel, содержащей 10–20 листов? Или долго вносить данные в файл, а потом случайно выйти из него без сохранения? Или сталкиваться с ситуацией, когда ваша книга Excel, вчера еще занимавшая меньше мегабайта, сегодня вдруг «распухла» до 10Мб?

Эта глава посвящена приемам и трюкам управления вашими рабочими книгами и листами внутри них, а именно:

- Как организовать **удобную навигацию** в книге по листам, когда их много.
- Как сделать **автоматическое оглавление книги** с гиперссылками.
- Как **сохранить каждый лист** в книге как отдельный файл и собрать листы из разных книг в одну.
- Как **отсортировать листы** в книге по алфавиту.
- Как бороться с **тяжелыми и медленными** файлами.
- Как автоматически делать **резервные копии** ценных файлов.



Удобная навигация по листам

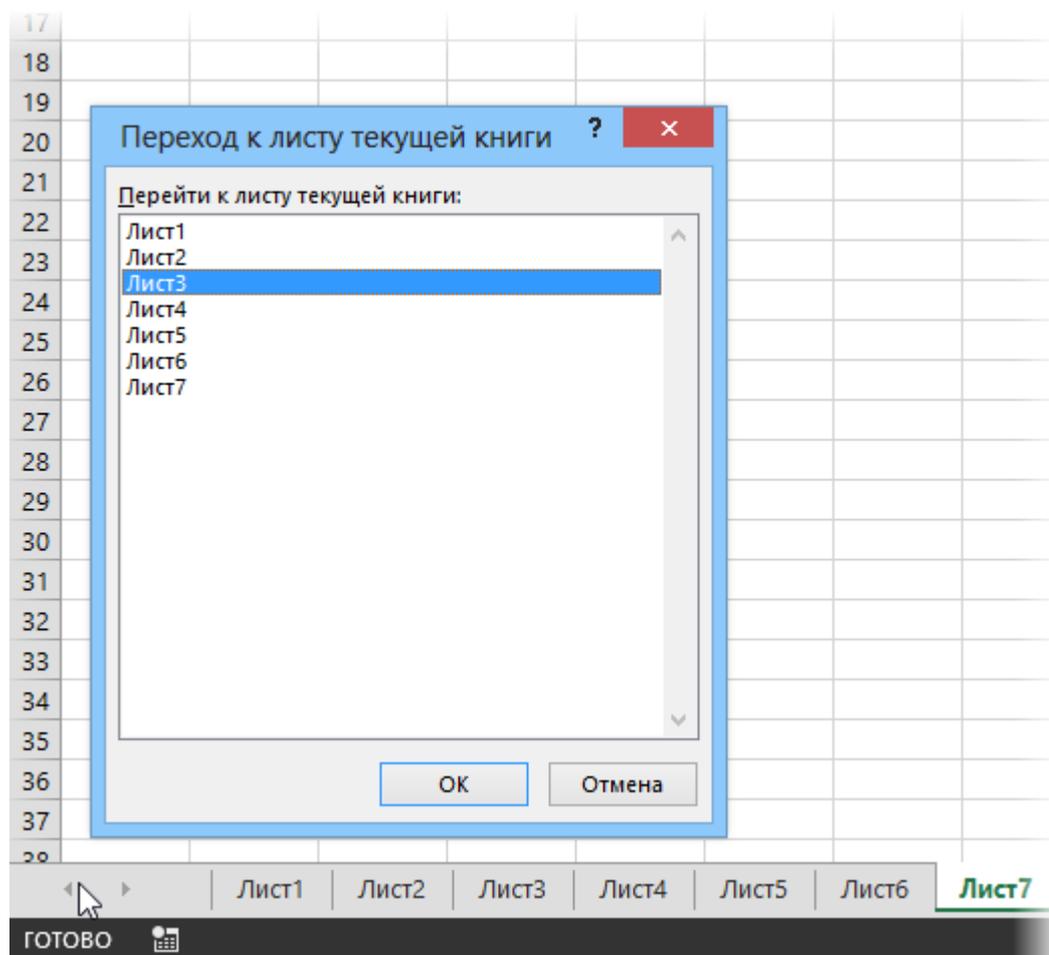
У вас бывают файлы, где много листов? Несколько десятков? Переход на нужный лист в такой книге может напрягать – пока найдешь нужный ярлычок листа, пока по нему щелкнешь...

Горячие клавиши

Сочетания **Ctrl+PgUp** и **Ctrl+PgDown** позволят вам быстро пролистывать вашу книгу вперед-назад.

Переход мышью

Просто щелкните правой кнопкой мыши по кнопкам прокрутки, находящимся слева от ярлычков листов (левый нижний угол окна Excel), и выберите нужный лист в появившемся окне:



Просто и изящно. Работает во всех версиях Excel.

Закладки

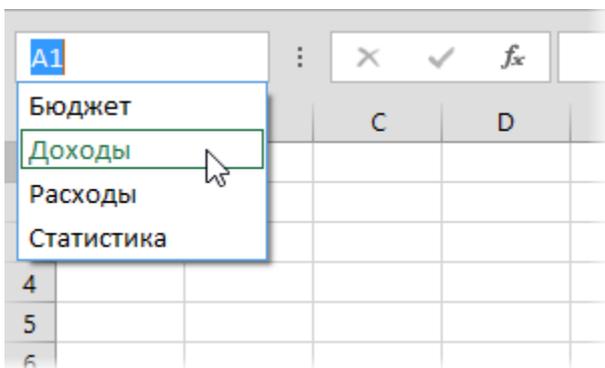
Если вам постоянно приходится "скакать" по листам вашей книги, переходя к одним и тем же ячейкам, например, для ввода данных или внесения корректировок, то имеет смысл сделать закладки на эти ячейки и переходить к ним быстро. Для этого:

1. Выделите ячейку или диапазон.
2. В левой части строки формул, где отображается адрес первой ячейки диапазона, введите вместо него имя закладки и нажмите **Enter**:

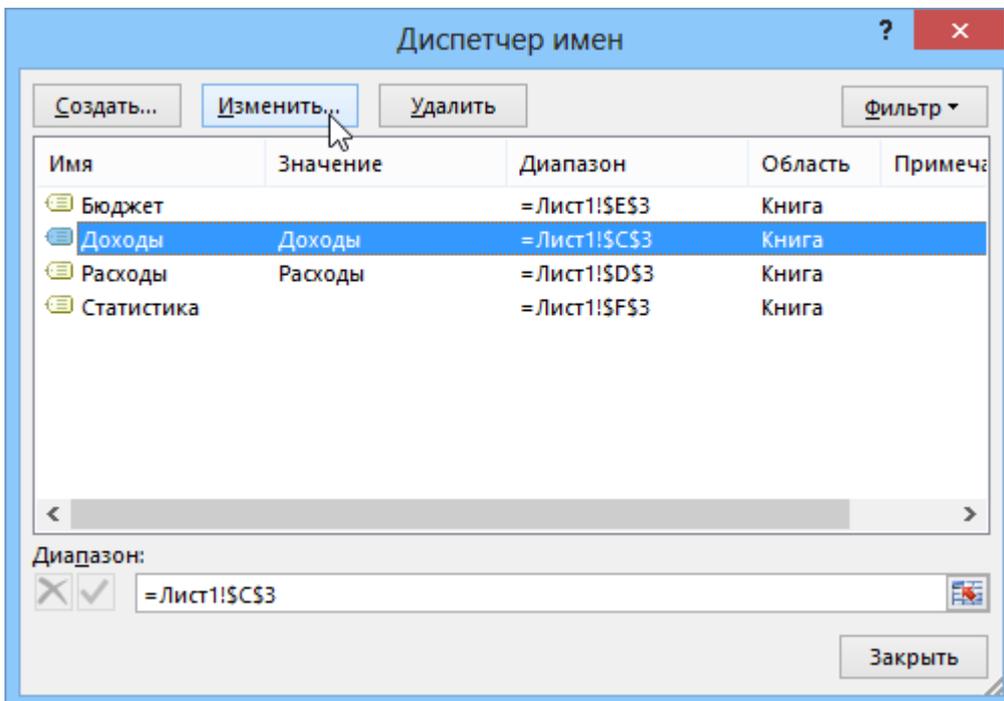
	A	B	C	D	E
1					
2		Отчет			
3			Доходы	Расходы	
4		янв	78744	89712	
5		фев	66032	53155	
6		мар	12285	67139	

Имя не должно содержать знаков препинания и пробелов, должно начинаться с буквы и быть не похожим на адрес ячейки (типа A5, XC90, C300 и т.д.).

Теперь к созданной закладке можно перейти с любого листа книги, используя раскрывающийся список **Имя (Name)** в левой части строки формул:



Для редактирования или удаления созданных закладок можно использовать кнопку **Диспетчер имен** на вкладке **Формулы (Formulas – Name Manager)**, которая откроет соответствующее диалоговое окно:



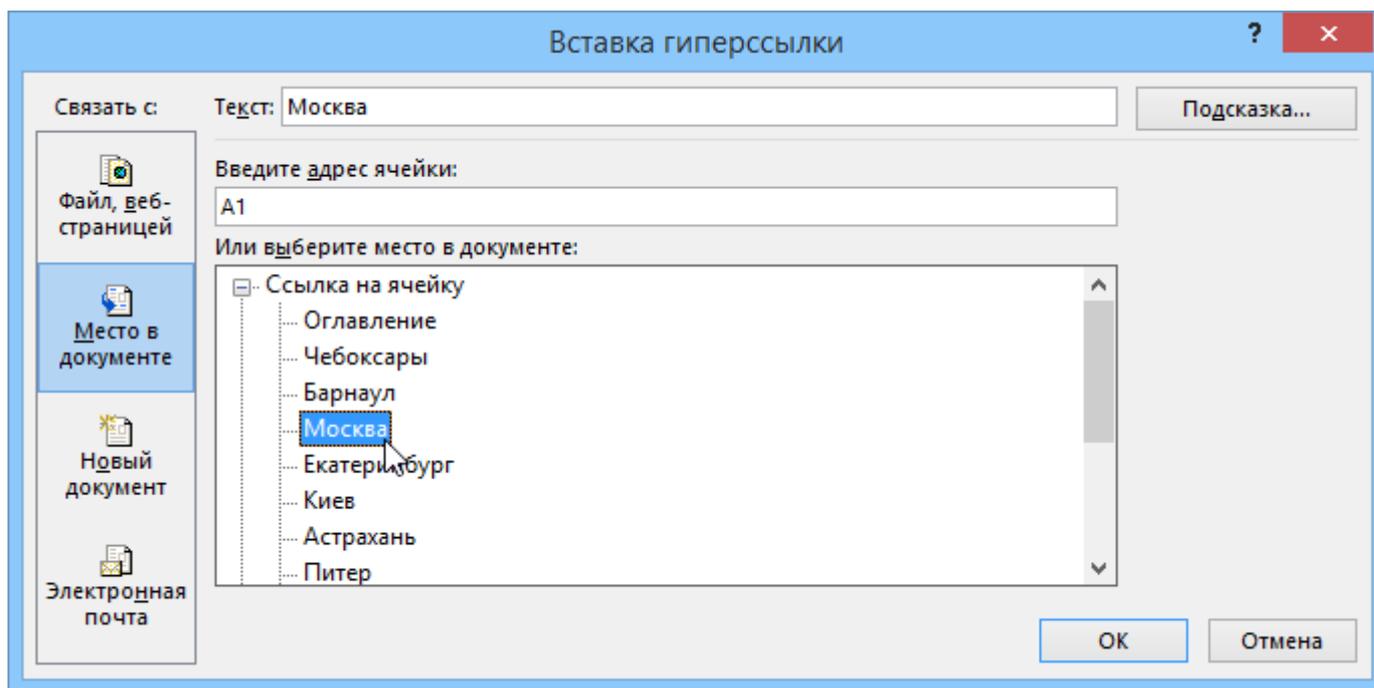
Оглавление книги

При большом количестве листов в рабочей книге навигация по ним может быть весьма утомительной. Одним из красивых способов ее решения является создание листа-оглавления с гиперссылками, ведущими на соответствующие листы книги:

	A	B
1	Чебоксары	
2	Барнаул	
3	Москва	
4	Екатеринбург	
5	Киев	
6	Астрахань	
7	Питер	
8	Самара	

Гиперссылки

Вставьте в книгу пустой лист и добавьте на него гиперссылки на нужные вам листы, используя команду **Вставка – Гиперссылка (Insert – Hyperlink)**. В открывшемся окне нужно выбрать слева опцию **Место в документе** и задать внешнее текстовое отображение и адрес ячейки, куда приведет ссылка:



Для удобства можно создать также и обратные ссылки на всех листах вашей книги, которые будут вести назад в оглавление. Чтобы не заниматься ручным созданием гиперссылок и копированием их потом на каждый лист, лучше использовать другой метод. Выделяем все листы в книге, куда хотим добавить обратную ссылку (для массового выделения листов можно использовать клавиши **Shift** и/или **Ctrl**), и в любую подходящую ячейку вводим функцию следующего вида:

```
=ГИПЕРССЫЛКА("#Оглавление!A1";"Назад в оглавление")
```

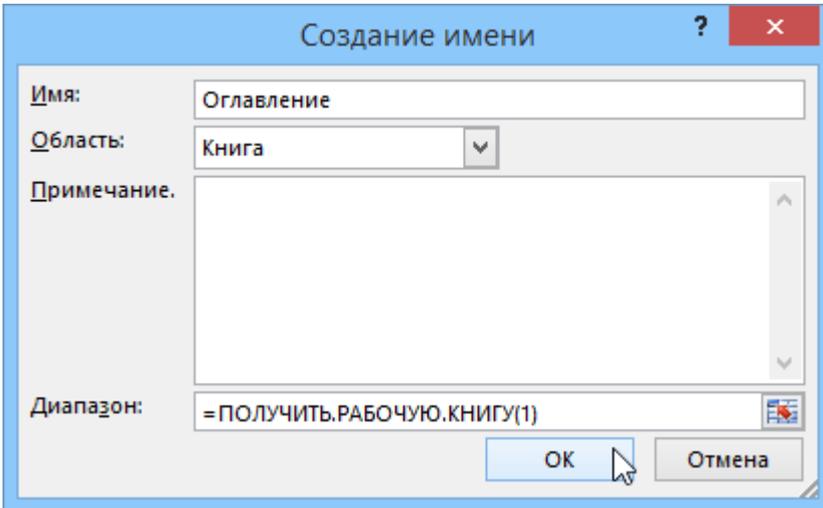
```
=HYPERLINK("#Оглавление!A1";"Назад в оглавление")
```

Эта функция создаст в текущей ячейке на всех выделенных листах гиперссылку с текстом "Назад в оглавление", щелчок по которой будет возвращать пользователя к листу *Оглавление*.

Динамическое оглавление с помощью формул

Это хоть и слегка экзотический, но весьма красивый и удобный способ создания автоматического листа оглавления вашей книги. Экзотический – потому что в нем используется недокументированная XLM-функция **ПОЛУЧИТЬ.РАБОЧУЮ.КНИГУ (GET.WORKBOOK)**, оставленная разработчиками для совместимости со старыми версиями Excel. Эта функция выгружает список всех листов текущей книги в заданную переменную, из которой мы потом можем их извлечь и использовать в нашем оглавлении.

Откройте **Диспетчер имен** на вкладке **Формулы (Formulas – Name Manager)** и создайте новый именованный диапазон с именем, допустим, *Оглавление*. В поле **Диапазон (Reference)** введите вот такую формулу:



Теперь в переменной *Оглавление* содержатся наши искомые имена. Чтобы извлечь их оттуда на лист, можно воспользоваться функцией **ИНДЕКС (INDEX)**, которая "выдергивает" элементы из массива по их номеру:

```
=ИНДЕКС(Оглавление; СТРОКА())
```

```
=INDEX(Оглавление; ROW())
```

Функция **СТРОКА (ROW)** выдает номер текущей строки и в данном случае нужна только для того, чтобы вручную не создавать отдельный столбец с порядковыми номерами извлекаемых элементов (1,2,3...). Таким образом, в ячейке A1 у нас получится имя первого листа, в A2 – имя второго и т.д.

	A	B	C	D	E	F	G
1							
2	[Оглавление.xlsm]Москва						
3	[Оглавление.xlsm]Екатеринбург						
4	[Оглавление.xlsm]Киев						
5	[Оглавление.xlsm]Барнаул						
6	[Оглавление.xlsm]Астрахань						
7	[Оглавление.xlsm]Питер						
8	[Оглавление.xlsm]Чебоксары						
9	[Оглавление.xlsm]Самара						
10	[Оглавление.xlsm]Воронеж						
11							
12							
13							

Formula bar: =ИНДЕКС(Оглавление;СТРОКА())

Sheet tabs: Оглавление | Москва | Екатеринбург | Киев | Барнаул | Астрахань | Питер

Неплохо. Однако, как можно заметить, функция выдает не только имя листа, но и полный путь к файлу, который нам не нужен. Чтобы его убрать, воспользуемся функциями **ЗАМЕНИТЬ (SUBST)** и **НАЙТИ (FIND)**, которые найдут символ закрывающей квадратной скобки (]) и заменят весь текст до этого символа включительно на пустую строку (""):

=ЗАМЕНИТЬ(ПОЛУЧИТЬ.РАБОЧУЮ.КНИГУ(1);1;НАЙТИ("]";ПОЛУЧИТЬ.РАБОЧУЮ.КНИГУ(1));"")

Теперь наш список листов будет выглядеть существенно лучше:

	A	B	C	D	E	F	G
1							
2	Москва						
3	Екатеринбург						
4	Киев						
5	Барнаул						
6	Астрахань						
7	Питер						
8	Чебоксары						
9	Самара						

Небольшая побочная трудность заключается в том, что наша формула в именованном диапазоне *Оглавление* будет пересчитываться только при вводе либо при принудительном пересчете книги нажатием на сочетание клавиш **Ctrl+Alt+F9**. Чтобы обойти этот неприятный момент, добавим к нашей формуле в **Диспетчере имен** небольшой хвост:

=ЗАМЕНИТЬ(ПОЛУЧИТЬ.РАБОЧУЮ.КНИГУ(1);1;НАЙТИ("]";ПОЛУЧИТЬ.РАБОЧУЮ.КНИГУ(1));"")&Т(ТДАТА())

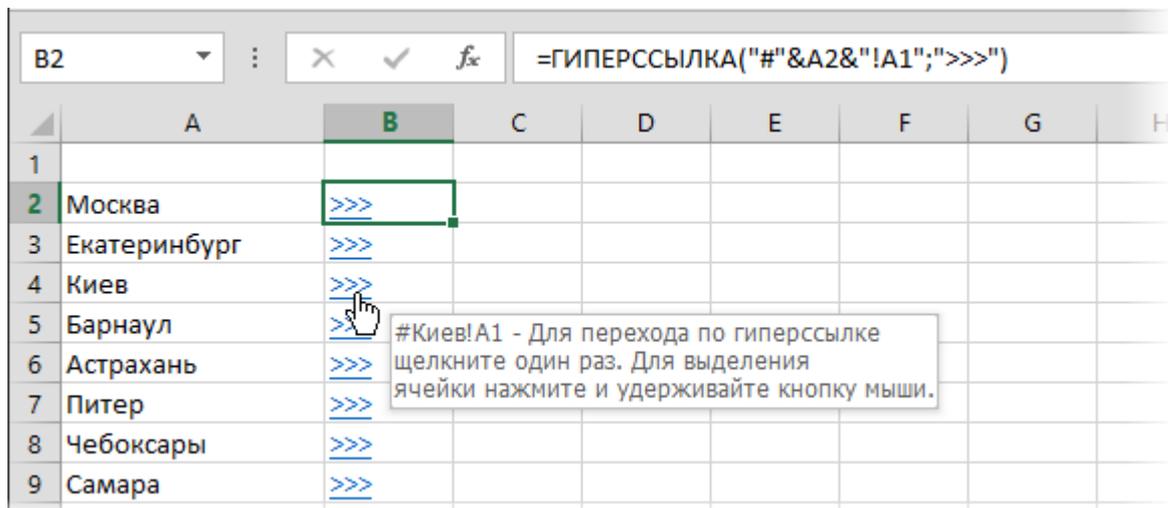
Функция **ТДАТА (NOW)** выдает текущую дату (с временем), а функция **Т** превращает эту дату в пустую текстовую строку, которая затем приклеивается к нашему имени листа с помощью оператора склейки (&). Т.е. имя листа, фактически, не меняется, но поскольку функция **ТДАТА** пересчитывается и выдает новое время и дату при любом изменении листа, то и остальная часть нашей формулы вынуждена будет заново пересчитаться тоже, и – как следствие – имена листов будут обновляться постоянно.

Для скрытия ошибок **#ССЫЛКА (#REF)**, которые будут появляться, если скопировать нашу формулу с функцией **ИНДЕКС** на большее количество ячеек, чем у нас есть листов, можно использовать функцию **ЕСЛИОШИБКА (IFERROR)**:

	A	B	C	D	E	F	G	H
1								
2	Москва							
3	Екатеринбург							
4	Киев							

Она перехватывает любые ошибки и заменяет их на пустую строку ("").

И, наконец, для добавления к именам листов "живых" гиперссылок для быстрой навигации можно использовать все ту же функцию **ГИПЕРССЫЛКА (HYPERLINK)**, которая будет формировать адрес для перехода из имени листа:



Макрос автоматического оглавления

Для решения задачи создания листа оглавления можно использовать и простой макрос.

Откройте редактор Visual Basic, нажав сочетание клавиш **Alt+F11** или выбрав на вкладке **Разработчик** кнопку **Visual Basic (Developer – Visual Basic)**. В открывшемся окне редактора создайте новый пустой модуль (меню **Insert – Module**) и введите туда текст этого макроса:

```
Sub TOC()
    Dim sheet As Worksheet
    Dim cell As Range

    For Each sheet In ActiveWorkbook.Worksheets
        Set cell = ActiveSheet.Cells(sheet.Index, 1)
        ActiveSheet.Hyperlinks.Add anchor:=cell, Address:="", _
            SubAddress:="" & sheet.Name & "" & "!A1"
        cell.Formula = sheet.Name
    Next
End Sub
```

Закройте редактор Visual Basic и вернитесь в Excel. Добавьте в книгу чистый лист и поместите его на первое место. Затем нажмите сочетание клавиш **Alt+F8** или нажмите кнопку **Макросы** на вкладке **Разработчик (Developer – Macros)**. Найдите в открывшемся окне созданный макрос *TOC* и запустите его на выполнение. Макрос создаст на первом листе книги список гиперссылок с названиями листов.

Если вы используете в книге скрытые листы для хранения вспомогательной информации, промежуточных вычислений и т.п. и не хотите, чтобы они попали в оглавление, то макрос придется чуть-чуть изменить, добавив проверку на видимость:

```
Sub TOC_visble_only()
    Dim sheet As Worksheet
    Dim cell As Range

    For Each sheet In ActiveWorkbook.Worksheets
        If sheet.Visible = True Then
            Set cell = ActiveSheet.Cells(sheet.Index, 1)
            ActiveSheet.Hyperlinks.Add anchor:=cell, Address:="", _
                SubAddress:="" & sheet.Name & "" & "!A1"
            cell.Formula = sheet.Name
        End If
    Next
End Sub
```

И, наконец, этот макрос можно поместить прямо в модуль листа и назначить событию его активации. Тогда при переходе на лист *Оглавление* он будет автоматически стирать его содержимое и генерировать его тут же еще раз. Это позволит автоматически отслеживать любые изменения в книге, добавление-удаление-переименование листов и т.д.

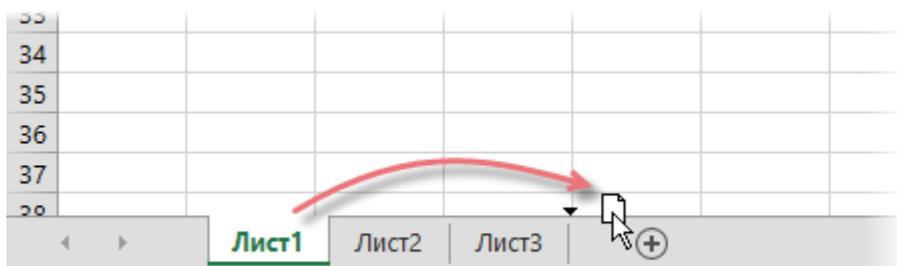
Чтобы это реализовать, щелкните правой кнопкой мыши по ярлычку листа *Оглавление* и выберите в контекстном меню команду **Исходный текст (Source Code)**. В появившееся окно нужно будет ввести наш макрос в слегка измененном варианте (как процедуру обработки события + удаление старых данных):

```
Private Sub Worksheet_Activate()  
    Dim sheet As Worksheet  
    Dim cell As Range  
  
    s=1  
    ActiveSheet.Cells.Clear    'очищаем лист от старого оглавления  
    With ActiveWorkbook  
        For Each sheet In ActiveWorkbook.Worksheets  
            If sheet.Visible = True Then  
                Set cell = ActiveSheet.Cells(s, 1)  
                s = s + 1  
                .ActiveSheet.Hyperlinks.Add anchor:=cell, Address:="", _  
                    SubAddress:="" & sheet.Name & "" & "!A1"  
                cell.Formula = sheet.Name  
            End If  
        Next  
    End With  
End Sub
```

Теперь в любой момент при открытии листа *Оглавление* наш макрос будет запускаться и создавать его заново – со всеми изменениями в структуре книги.

Быстрое копирование листов

Самый быстрый способ скопировать лист – это перетащить его ярлычок, удерживая клавишу **Ctrl**:

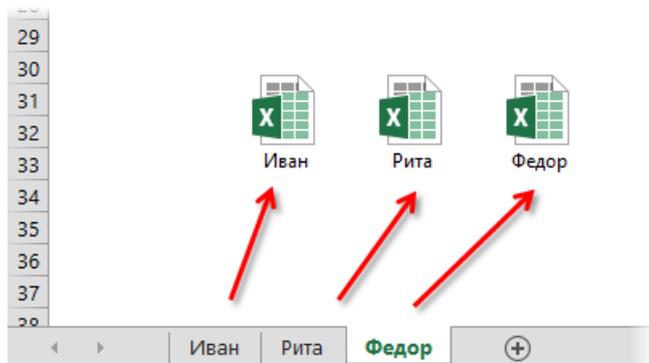


Если до перетаскивания сначала выделить несколько листов (удерживая клавиши **Ctrl** или **Shift**), то можно скопировать сразу несколько листов.

Сохранение выбранных листов как отдельных файлов

Предположим, у нас есть книга Excel, которую нужно "разобрать", т.е. сохранить каждый лист как отдельный файл для дальнейшего использования.

Примеров подобного из реальной жизни можно привести массу. Например, файл-отчет с листами-сотрудниками нужно разделить на отдельные книги по листам, чтобы передать затем данные каждому человеку отдельно и т.д.



Если делать эту процедуру вручную, то придется для каждого листа выполнить немаленькую цепочку действий:

1. выбрать лист;
2. правой кнопкой по ярлычку листа – выбрать **Переместить или скопировать (Move or Copy)**;
3. в открывшемся окне указать отдельный предварительно созданный пустой файл;
4. не забыть поставить флажок **Создать копию** (иначе будет перенос);
5. нажать **ОК**;
6. сохранить созданный файл;
7. повторить все действия для каждого листа.

Уф! Гораздо проще использовать простые макросы для автоматизации этих скучных действий.

Простое разделение листов по файлам

Нажмите сочетание **Alt+F11** или выберите на вкладке **Разработчик** кнопку **Редактор Visual Basic (Developer – Visual Basic Editor)**, вставьте новый модуль через меню **Insert – Module** и введите туда текст этого макроса:

```
Sub SplitSheets1()
    Dim s As Worksheet
    For Each s In ActiveWorkbook.Worksheets
        s.Copy 'проходим по всем листам в активной книге
    Next 'копируем каждый лист в новый файл
End Sub
```

Если теперь выйти из редактора Visual Basic и вернуться в Excel, а затем запустить наш макрос с помощью диалогового окна по нажатию **Alt+F8** или с помощью кнопки **Макросы** на вкладке **Разработчик (Developer – Macros)**, то все листы из текущей книги будут разбиты по отдельным новым созданным книгам.

Разделение с сохранением

При необходимости можно созданные книги сразу же сохранять под именами листов. Для этого макрос придется немного изменить, добавив команду сохранения в цикл:

```
Sub SplitSheets2()
    Dim s As Worksheet
    Dim wb as Workbook
    Set wb = ActiveWorkbook
    For Each s In wb.Worksheets
        s.Copy 'проходим во всем листам книги
        ActiveWorkbook.SaveAs wb.Path & "\" & s.Name & ".xls" 'сохраняем лист как новый файл
    Next 'сохраняем файл
End Sub
```

Этот макрос сохраняет новые книги-листы в ту же папку, где лежал исходный файл. При необходимости сохранения в другое место замените `wb.Path` на свой путь в кавычках, например "D:\Отчеты\2012" и т.п. Естественно, и расширение `xls` можно заменить на свое (`xlsx` или `xlsm`) в зависимости от вашей версии Excel.

Сохранение в новые книги только выделенных листов

Если вы хотите раскидать по файлам не все листы в вашей книге, а только некоторые, то макрос придется немного изменить, чтобы он экспортировал только выделенные листы:

```
Sub SplitSheets3()
    Dim AW As Window
    Set AW = ActiveWindow
    For Each s In AW.SelectedSheets
        Set TempWindow = AW.NewWindow      'создаем отдельное временное окно
        s.Copy                               'копируем туда лист из выделенного диапазона
        TempWindow.Close                    'закрываем временное окно
    Next
End Sub
```

Выделите нужные вам листы в книге, удерживая на клавиатуре клавиши **Ctrl** или **Shift**, и запустите приведенный выше макрос.

Сохранение только выделенных листов в новый файл

Во всех описанных выше способах каждый лист сохранялся в свой отдельный файл. Если же вы хотите сохранить в один (!) отдельный новый файл сразу группу выделенных предварительно листов, то нам потребуется слегка видоизменить наш макрос:

```
Sub SplitSheets4()
    Dim CurW As Window
    Dim TempW As Window
    Set CurW = ActiveWindow
    Set TempW = ActiveWorkbook.NewWindow
    CurW.SelectedSheets.Copy
    TempW.Close
End Sub
```

Сборка листов из разных книг в одну

Предположим, имеется куча книг Excel, все листы из которых надо объединить в один файл. Копировать руками долго и мучительно, поэтому имеет смысл использовать несложный макрос.

Открываем книгу, куда хотим собрать листы из других файлов, входим в редактор Visual Basic (**Alt+F11**), добавляем новый пустой модуль (в меню **Insert – Module**) и вставляем туда текст вот такого макроса:

```
Sub CombineWorkbooks()  
    Dim FilesToOpen  
    Dim x As Integer  
  
    Application.ScreenUpdating = False 'отключаем обновление экрана для скорости  
  
    'вызываем диалог выбора файлов для импорта  
    FilesToOpen = Application.GetOpenFilename _  
        (FileFilter:="All files (*.*), *.*", _  
        MultiSelect:=True, Title:="Files to Merge")  
  
    If TypeName(FilesToOpen) = "Boolean" Then  
        MsgBox "Не выбрано ни одного файла!"  
        Exit Sub  
    End If  
  
    'проходим по всем выбранным файлам  
    x = 1  
    While x <= UBound(FilesToOpen)  
        Set importWB = Workbooks.Open(FileName:=FilesToOpen(x))  
        Sheets().Copy After:=ThisWorkbook.Sheets(ThisWorkbook.Sheets.Count)  
        importWB.Close savechanges:=False  
        x = x + 1  
    Wend  
  
    Application.ScreenUpdating = True  
End Sub
```

После этого можно вернуться в Excel и запустить созданный макрос на вкладке **Разработчик – Макросы (Developer – Macros)** или нажав **Alt+F8**. Отобразится диалоговое окно открытия файла, где необходимо указать один или несколько (удерживая **Ctrl** или **Shift**) файлов, листы из которых надо добавить к текущей книге.

Сортировка листов

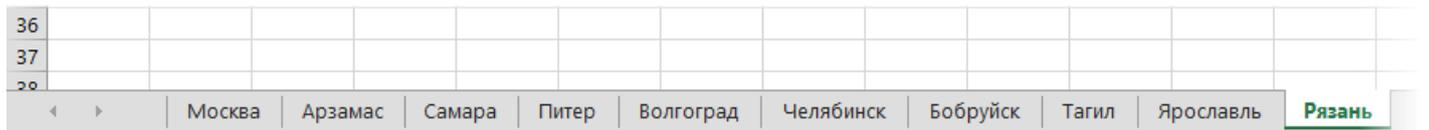
Если количество листов в вашей книге приближается к нескольким десяткам, то рано или поздно возникнет желание отсортировать листы, разложив их по порядку для удобства навигации. Стандартные средства Excel не позволяют сделать этого, но можно использовать простой макрос, который реализует такую сортировку.

Откройте редактор Visual Basic с помощью сочетания клавиш **Alt+F11**, вставьте новый модуль (меню **Insert – Module**) и вставьте туда код этого макроса:

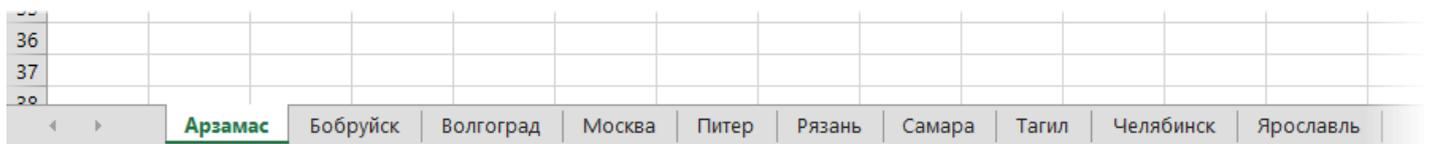
```
Sub SortSheets()
    Dim I As Integer, J As Integer

    For I = 1 To Sheets.Count - 1
        For J = I + 1 To Sheets.Count
            If UCase(Sheets(I).Name) > UCase(Sheets(J).Name) Then
                Sheets(J).Move Before:=Sheets(I)
            End If
        Next J
    Next I
End Sub
```

Теперь этот макрос можно запустить на вкладке **Разработчик – Макросы (Developer – Macros)** или нажав **Alt+F8** и выбрав команду **Выполнить (Run)**: он быстро отсортирует все листы в текущей книге по алфавиту (по возрастанию), т.е. сделает из:



...ВОТ ЭТО:



Оптимизация быстродействия

Если в один прекрасный момент вы осознаете, что ваш основной рабочий файл в Excel разбух до нескольких десятков мегабайт и во время работы с ним вы то и дело тренируете свое терпение, то попробуйте пробежаться по описанным ниже пунктам – возможно, один или несколько из них укоротят вашего "переростка" до вменяемых размеров и ускорят его.

Новые форматы файлов Excel 2007–2013

В новых версиях Excel 2007–2013 введены новые форматы сохранения книг, использование которых заметно облегчает жизнь:

- **XLSX** – по сути является заархивированным Winzip'ом файлом формата XML. Размер файлов в таком формате по сравнению с форматом XLS старых версий Excel 97–2003 меньше в среднем в 5–7 раз!
- **XLSM** – то же самое, но с поддержкой макросов.
- **XLSB** – двоичный формат, скомпилированный в виде бинарного кода XML. С другими приложениями, кроме Excel, совместимости нет (а она нужна?), но зато скорость доступа к данным в файле – максимальная, а его размер – минимален.

Везде, где можно, лучше использовать новые форматы вместо старых. Они компактнее, быстрее и надежнее хранят данные (устойчивее к повреждениям). Для преобразования книг старых форматов в новые достаточно открыть их в Excel и воспользоваться командой **Файл – Преобразовать (File – Convert)**.

Используемый диапазон

Если ваша таблица занимает 5 на 5 ячеек, то это отнюдь не означает, что Excel запоминает при сохранении этого файла только 25 ячеек с данными. Если вы в прошлом использовали какие-либо ячейки на этом листе, то они автоматически включаются в используемый диапазон (used range), который и запоминается при сохранении книги. Проблема в том, что при очистке используемых ячеек Excel далеко не всегда автоматически исключает их из используемого диапазона, т.е. начинает запоминать в файле больше данных, чем реально имеется.

Проверить это просто – нажмите на клавиатуре сочетание клавиш **Ctrl+End** и посмотрите, куда переместится активная ячейка. Если она прыгнет на фактическую последнюю ячейку с данными на листе – отлично. А если вдруг ускачет сильно правее и/или ниже – дело плохо. Лечится это тем не менее достаточно легко:

1. Выделите первую пустую строку под вашей таблицей.
2. Нажмите сочетание клавиш **Ctrl+Shift+стрелка вниз** – выделятся все пустые строки до конца листа.
3. Удалите их, нажав на клавиатуре **Ctrl+знак минус** или выбрав на вкладке **Главная – Удалить – Удалить строки с листа (Home – Delete – Delete rows)**.
4. Повторите то же самое со столбцами.
5. Повторите все вышеописанные процедуры на каждом листе, где при нажатии на **Ctrl+End** активная ячейка перемещается не на фактическую последнюю ячейку с данными, а "в пустоту" ниже и/или правее.
6. Сохраните файл, чтобы изменения вступили в силу.

В особо тяжелых случаях после такой процедуры мне удавалось уменьшить размер файла в 2–3 раза.

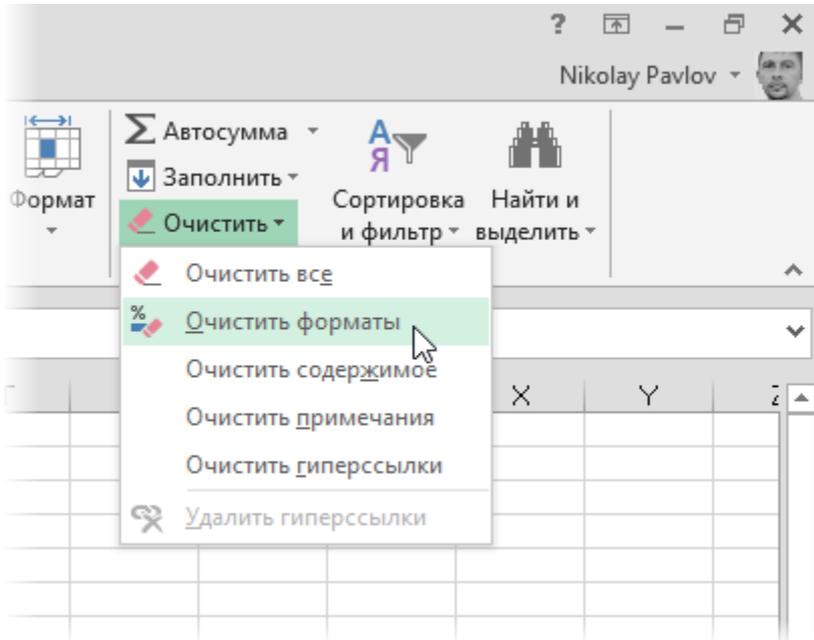
Если подобную процедуру приходится делать часто или в книге очень много листов, то имеет смысл воспользоваться для такой "зачистки" простым макросом. Нажмите сочетание клавиш **Alt+F11** или кнопку **Visual Basic** на вкладке **Разработчик (Developer – Visual Basic)**. В открывшемся окне создайте новый модуль с помощью меню **Insert – Module** и введите туда текст нашего макроса:

```
Sub ClearBook()  
    For Each wsh In ActiveWorkbook.Worksheets  
        wsh.UsedRange  
    Next wsh  
End Sub
```

Теперь макрос можно запустить прямо из редактора Visual Basic с помощью клавиши **F5** или нажав на вкладке **Разработчик** кнопку **Макросы** и затем **Выполнить (Developer – Macros – Run)**.

Форматирование

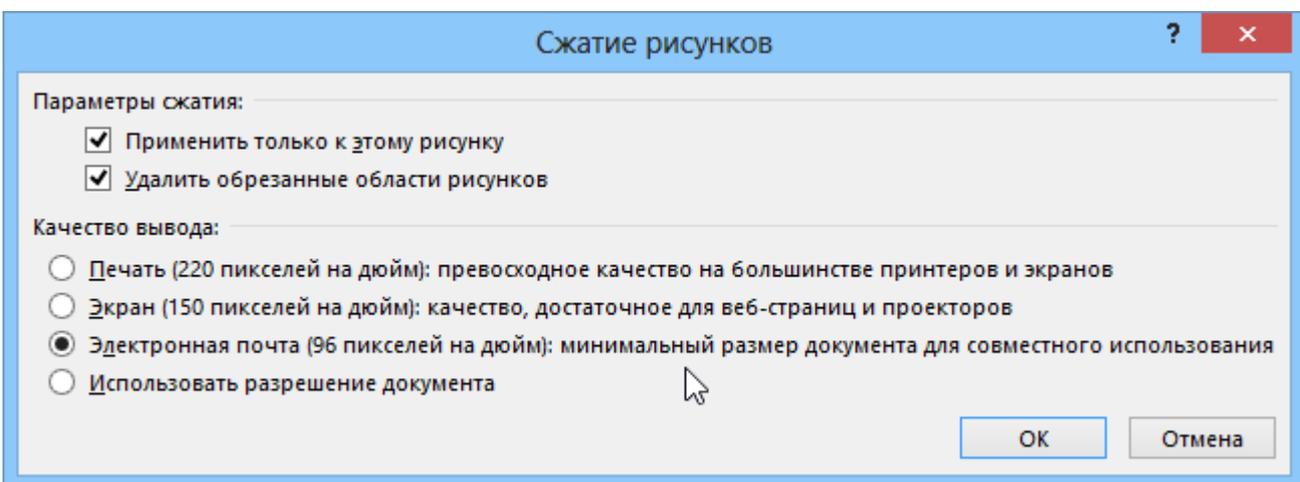
Сложное многоцветное форматирование и, особенно, условное форматирование заметно увеличивают размер файла. Оставьте только самое необходимое, не изощряйтесь. Особенно в тех таблицах, которые, кроме вас, никто не видит. Для удаления форматов выделите ячейки и выберите кнопку **Очистить** на вкладке **Главная (Home – Clear – Clear formats)**:



Особенно "загружают" файл отформатированные целиком строки и столбцы. Т.к. размер листа в последних версиях Excel сильно увеличен (>1 млн строк и >16 тыс. столбцов), то для запоминания и обработки подобного форматирования нужно много ресурсов.

Картинки

Само собой, тоже значительно увеличивают размер файла, особенно когда их много – в каталоге продукции и т.п. Советую сжимать их, уменьшая разрешение до 90 точек на дюйм. На экране по качеству это практически не заметно, а размер файла уменьшает прилично. Для сжатия воспользуйтесь кнопкой **Сжать рисунки (Compress Pictures)** на вкладке **Формат (Format)**. Эта кнопка открывает вот такое диалоговое окно:



Если отключить первый флажок **Применить только к этому рисунку** в этом окне, то сжатие будет применено ко всем рисункам в документе.

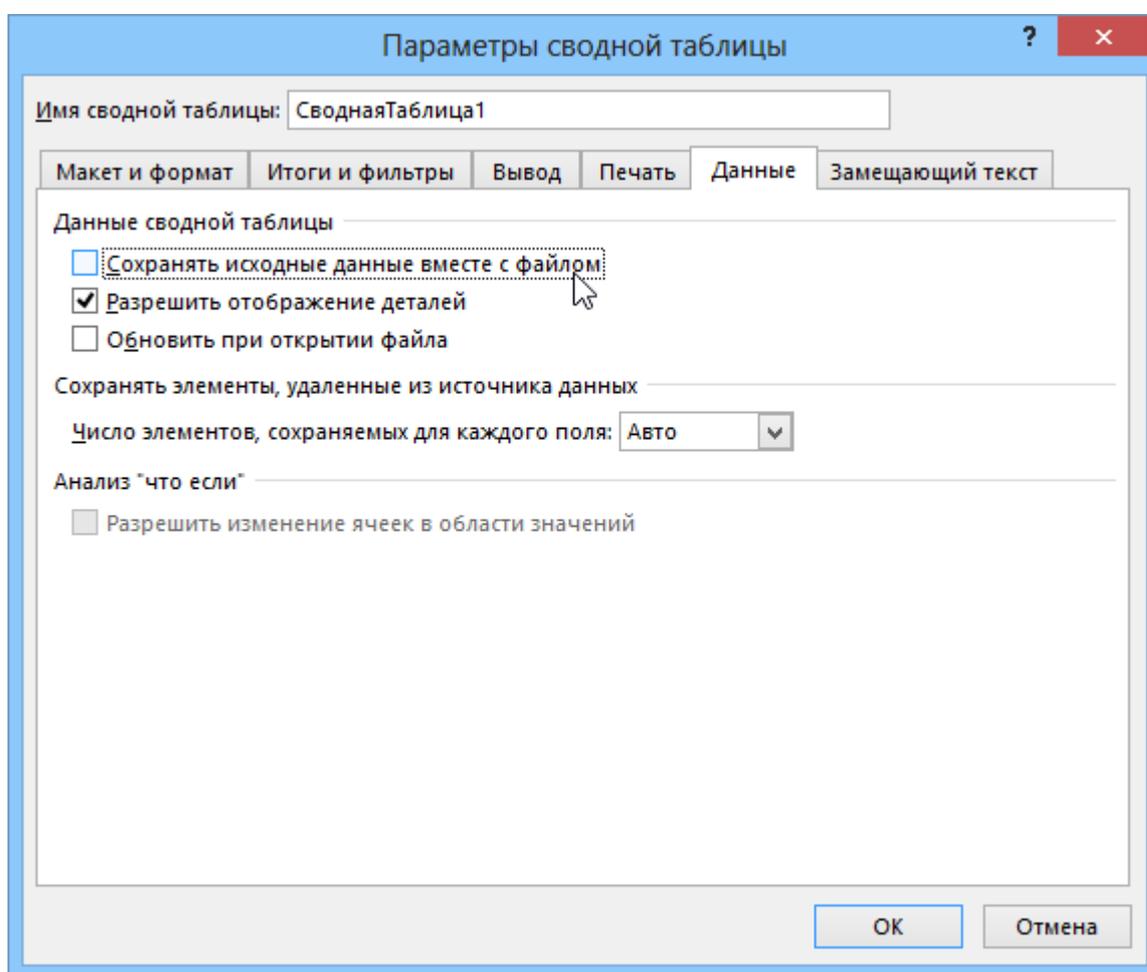
Если хотите просто удалить сразу все картинки на листе, то можно на вкладке **Главная** выбрать команду **Найти и выделить – Выделение группы ячеек (Home – Find & Select – Go To Special)** и выбрать в открывшемся окне опцию **Объекты (Objects)**. После этого их всех можно удалить клавишей **Delete**.

Примечания

Некоторые файлы могут содержать большое количество текста или даже картинки в примечаниях к ячейкам. Если эти примечания не содержат полезной для вас информации, то их можно легко удалить с помощью вышеприведенной команды: **Главная – Очистить – Примечания (Home – Clear – Comments)**.

Кэш сводных таблиц

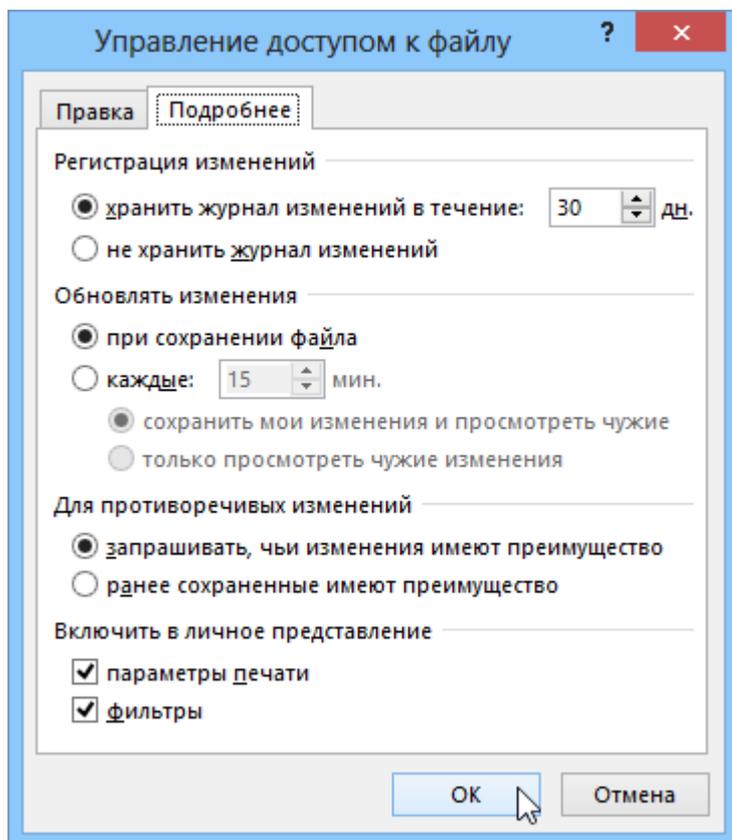
По умолчанию Excel сохраняет данные для расчета сводной таблицы (pivot cache) внутри файла. Можно отказаться от этой возможности, заметно сократив размер файла, но увеличив время на обновление сводной таблицы при следующем открытии книги. Щелкните правой кнопкой мыши по сводной таблице и выберите команду **Свойства таблицы (Pivot Table Properties)** – вкладка **Данные (Data)** – снять флажок **Сохранять исходные данные вместе с файлом (Save source data with file)**:



Журнал изменений (логи)

Если в вашем файле включен общий доступ на вкладке **Рецензирование – Доступ к книге (Review – Share Workbook)**, то внутри вашего файла Excel на специальном скрытом листе начинает сохраняться вся история изменений документа: кто, когда и как менял ячейки всех листов. По умолчанию такой журнал сохраняет данные изменений за последние 30 дней, т.е. при активной работе с файлом может запросто занимать несколько мегабайт.

Мораль – не используйте общий доступ без необходимости или сократите количество дней хранения данных журнала, используя вторую вкладку **Подробнее (Advanced)** в окне **Доступ к книге**. Там можно найти параметр **Хранить журнал изменений в течение N дней (Keep change history for N days)** или совсем отключить его:



Макросы и формы на VBA

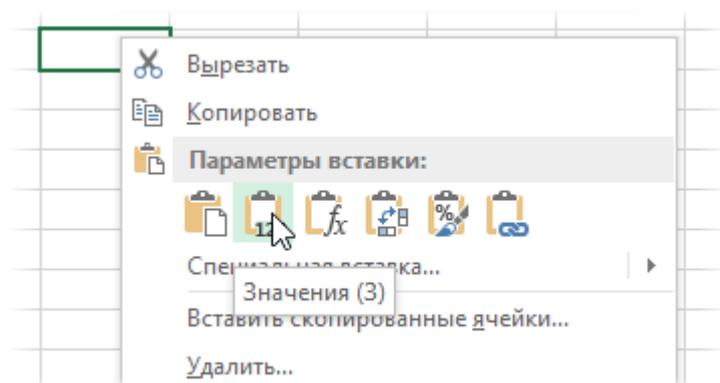
Большие макросы на Visual Basic и особенно пользовательские формы с внедренной графикой могут весьма заметно утяжелять вашу книгу. Для удаления можно просто сохранить файл в формате XLSX (без поддержки макросов) – все макросы и формы будут удалены автоматически.

Ручной пересчет формул

Можно избежать томительных пауз во время редактирования тяжелой книги с большим количеством вычислений, если отключить автоматический пересчет формул на вкладке **Формулы – Параметры вычислений – Вручную (Formulas – Calculation Options – Manual)**. Microsoft Excel перестанет пересчитывать формулы "на лету" при изменении любой ячейки (как он это обычно делает), а станет это производить только при нажатии на клавишу **F9** (пересчет всей книги) или **Shift+F9** (пересчет текущего листа).

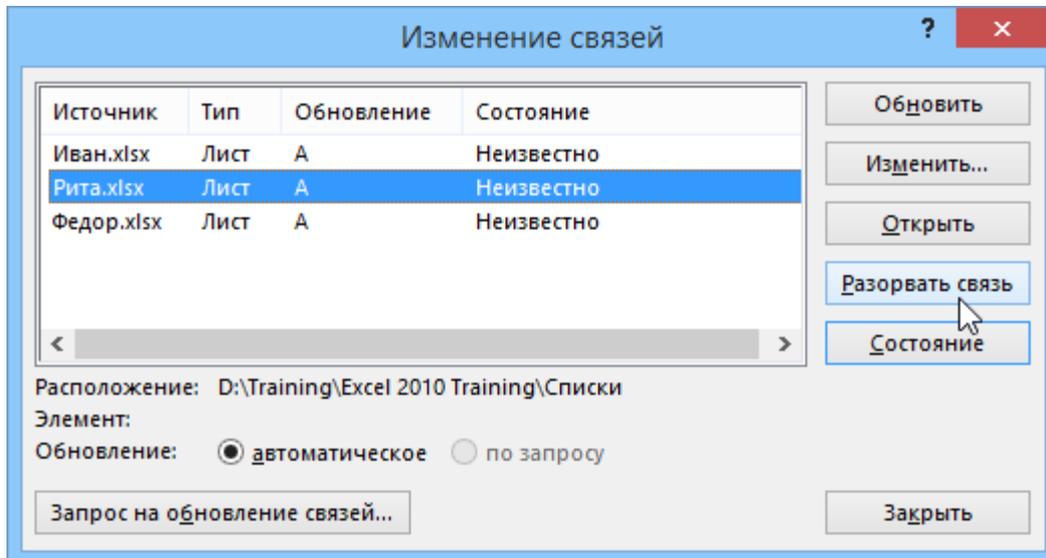
Замена формул на константы

Радикальный, но эффективный способ – удалить все формулы в книге или на листе, заменив их значениями. Т.е. выделить все ячейки на листе, скопировать их (правой кнопкой мыши или **Ctrl+C**) и туда же обратно вставить, но уже значения, используя **Специальную вставку** в контекстном меню правой кнопкой мыши или сочетание клавиш **Ctrl+Alt+V**:



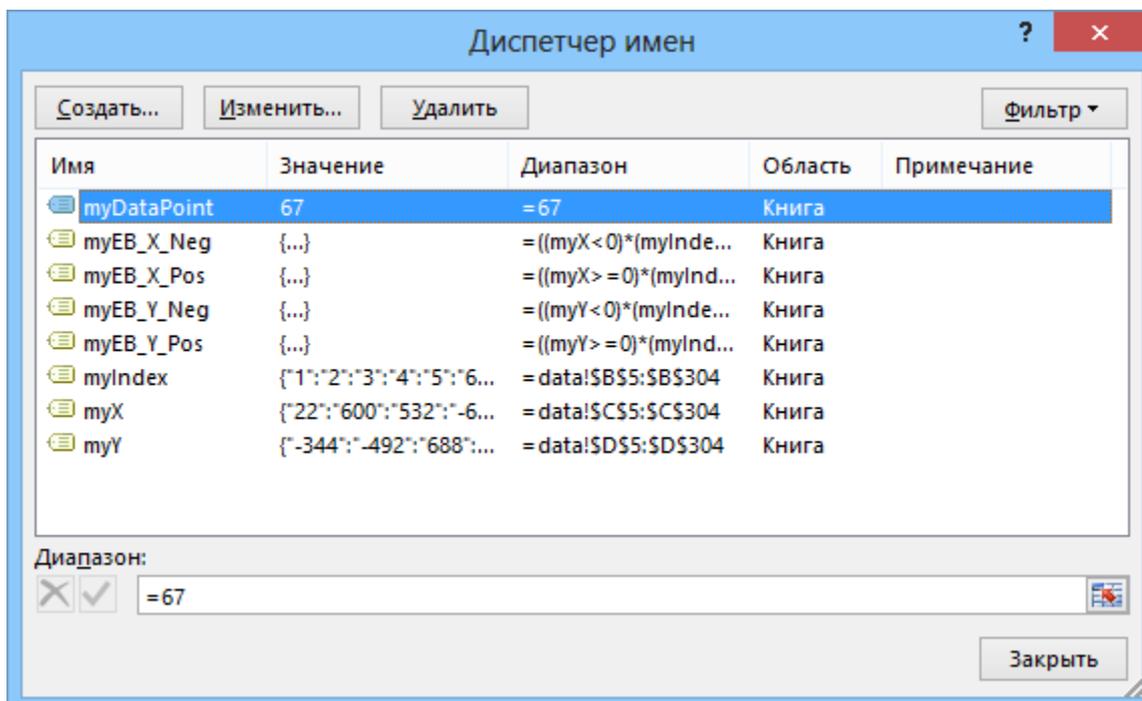
Внешние ссылки

Если ваша книга ссылается в формулах на другие файлы, то это однозначно не улучшает ее быстродействия. Такие книги, как правило, тяжелее, медленнее и ошибки обновления в них возникают нередко. По возможности заменяйте везде, где это допустимо, внешние ссылки на другие файлы на константы, используя, например, специальную вставку значений или команду **Изменить связи** на вкладке **Данные (Data – Edit Links)**. В появившемся диалоговом окне можно увидеть все связи с другими книгами и разорвать их, нажав на кнопку **Разорвать связь (Break Link)**:



Именованные диапазоны

Если в вашем файле используются именованные диапазоны (особенно с формулами, динамические или получаемые при фильтрации), то имеет смысл от них отказаться в пользу быстродействия при пересчете и экономии размера книги. Посмотреть список имеющихся диапазонов можно, нажав **Ctrl+F3** или открыв окно **Диспетчера имен (Name Manager)** на вкладке **Формулы (Formulas)**:



Условное форматирование с формулами

Формулы, замедляющие работу Excel, могут содержаться не только в ячейках, но и прятаться в условном форматировании, когда оформление ячеек меняется автоматически при изменении данных. Чтобы очистить

избыточные форматы, выделите ячейки и выберите на вкладке **Главная – Условное форматирование – Удалить правила (Home – Conditional Formatting – Delete Rules)**.

Получение списка файлов в папке

Есть много ситуаций, когда вам неплохо было бы иметь на листе Excel список всех файлов из определенной папки компьютера. Может быть, вы решили навести, наконец, порядок в своей коллекции фотографий или аудиозаписей mp3. Или вам нужно отслеживать документооборот по вашим клиентам. Или получить список всех файлов по проекту для ТЗ.

Решить подобную задачу можно двумя способами.

Способ 1. Формулы

В наследство от старых версий новому Excel 2013 досталось несколько так называемых макро XLM функций. Они присутствуют в Excel неявно, их не найти в **Мастере функций** или в справке, но если ввести их с клавиатуры, то они работают!

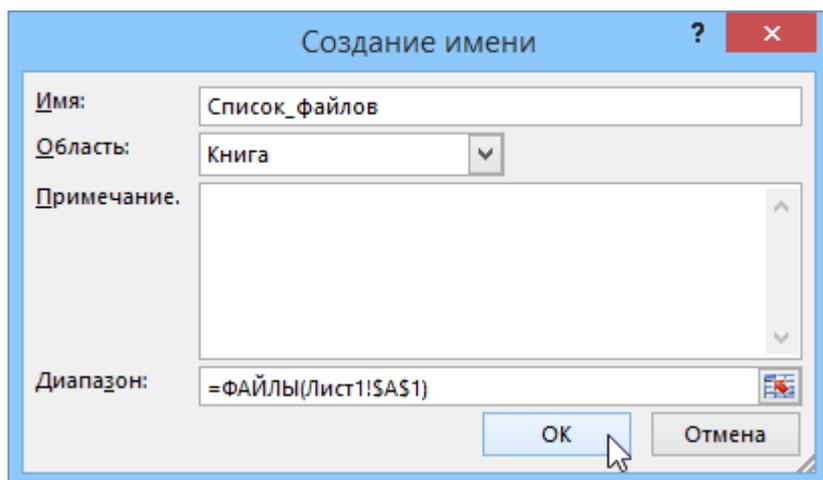
Одна из таких функций – функция **ФАЙЛЫ (FILES)**. Она выдает одномерный массив (столбец) файлов из указанной ей в качестве аргумента папки. Чтобы ее использовать, введем в ячейку A1 чистого листа путь к нужной папке:

	A	B	C
1	D:\Music\ACDC*.*		
2			
3			
4			
5			

Обратите внимание, что путь должен заканчиваться маской поиска файлов вида:

- *.* – все файлы;
- *.xl* – все файлы Excel (xls,xlsx,xlsm,xlsb и т.д.);
- *.exe – все исполняемые файлы (программы);
- **Отчет*.docx** – все документы Microsoft Word, чьи имена файлов начинаются со слова «Отчет» и т.д.

Теперь откроем **Диспетчер имен** на вкладке **Формулы (Formulas – Name Manager)**, создадим новое имя *Список_файлов* и введем в поле **Диапазон (Reference)** следующую формулу:



=ФАЙЛЫ(Лист1!\$A\$1)

или в англоязычной версии =FILES(Лист1!\$A\$1)

Теперь в созданной нами переменной *Список_файлов* уже содержатся нужные имена файлов, осталось их оттуда извлечь. Для этого используем в ячейке A3 следующую формулу и копируем ее вниз:

	A	B	C	D	E	F
1	D:\Music\ACDC*.*					
2						
3	ACDC - 01 - T.N.T..mp3					
4	ACDC - 02 - Problem Child.mp3					
5	ACDC - 03 - Whole Lotta Rosie.mp3					
6	ACDC - 04 - Sin City.mp3					
7	ACDC - 05 - Highway To Hell.mp3					

=ЕСЛИОШИБКА(ИНДЕКС(Список_файлов;СТРОКА()-2);"")

=IFERROR(INDEX(Список_файлов;ROW()-2);"")

Функция **ИНДЕКС (INDEX)** извлекает из массива элемент по номеру, например, формула

=**ИНДЕКС(Список_файлов; 3)** выдаст имя третьего по счету файла в папке.

Чтобы не делать отдельный столбец с номерами (1, 2, 3...), мы задаем номер с помощью функции **СТРОКА (ROW)**, которая выдает порядковый номер текущей строки, из которого мы вычитаем 2, т.к. первая ячейка, куда выводится имя, в нашем случае А3.

И, наконец, функция **ЕСЛИОШИБКА (IFERROR)** нужна, чтобы спрятать ошибку **#ССЫЛКА (#REF)**, которая возникает, когда мы протягиваем нашу формулу "с запасом", т.е. на больший по размеру диапазон, чем количество имеющихся у нас файлов.

Способ 2. Макрос

Список файлов из заданной папки можно получить и с помощью макроса (весьма сложного, правда). Зато макрос может выдать вам содержимое и вложенных папок, что невозможно в предыдущем способе, а также много другой полезной информации о файлах (даты создания и изменения, размер файлов и т.д.).

Откройте редактор Visual Basic сочетанием клавиш **Alt+F11**, добавьте новый пустой модуль (в меню **Insert – Module**) и введите туда текст вот такого макроса:

```
Sub FileList()
    Dim V As String
    Dim BrowseFolder As String

    With Application.FileDialog(msoFileDialogFolderPicker)
        .Title = "Выберите папку или диск"
        .Show
        On Error Resume Next
        Err.Clear
        V = .SelectedItems(1)
        If Err.Number <> 0 Then
            MsgBox "Вы ничего не выбрали!"
            Exit Sub
        End If
    End With
    BrowseFolder = CStr(V)

    'добавляем лист и выводим на него шапку таблицы
    ActiveWorkbook.Sheets.Add
    With Range("A1:E1")
        .Font.Bold = True
        .Font.Size = 12
    End With
    Range("A1").Value = "Имя файла"
    Range("B1").Value = "Путь"
    Range("C1").Value = "Размер"
    Range("D1").Value = "Дата создания"
    Range("E1").Value = "Дата изменения"
```

```

'вызываем процедуру вывода списка файлов
'измените True на False, если не нужно выводить файлы из вложенных папок
ListFilesInFolder BrowseFolder, True
End Sub

Private Sub ListFilesInFolder(ByVal SourceFolderName As String, ByVal IncludeSubfolders As Boolean)
'процедура вывода списка файлов в папке

Dim FSO As Object
Dim SourceFolder As Object
Dim SubFolder As Object
Dim FileItem As Object
Dim r As Long

Set FSO = CreateObject("Scripting.FileSystemObject")
Set SourceFolder = FSO.getfolder(SourceFolderName)

r = Range("A65536").End(xlUp).Row + 1 'находим первую пустую строку
'выводим данные по файлу
For Each FileItem In SourceFolder.Files
Cells(r, 1).Formula = FileItem.Name
Cells(r, 2).Formula = FileItem.Path
Cells(r, 3).Formula = FileItem.Size
Cells(r, 4).Formula = FileItem.DateCreated
Cells(r, 5).Formula = FileItem.DateLastModified
r = r + 1
X = SourceFolder.Path
Next FileItem

'вызываем процедуру повторно для каждой вложенной папки
If IncludeSubfolders Then
For Each SubFolder In SourceFolder.SubFolders
ListFilesInFolder SubFolder.Path, True
Next SubFolder
End If

Columns("A:E").AutoFit

Set FileItem = Nothing
Set SourceFolder = Nothing
Set FSO = Nothing

End Sub

```

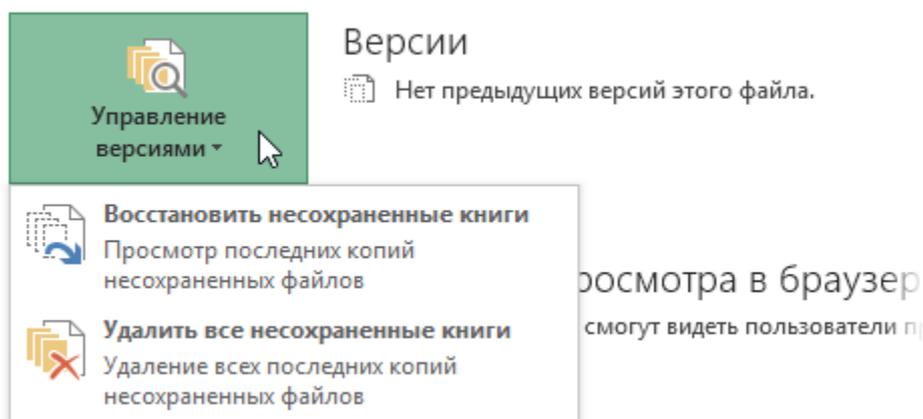
Если теперь выйти из редактора Visual Basic обратно в Excel и запустить созданный макрос с помощью кнопки **Макросы** на вкладке **Разработчик (Developer – Macros)** или с помощью сочетания клавиш **Alt+F8**, то мы получим диалоговое окно с предложением выделить папку или диск. После выбора список всех файлов из указанного расположения и всех вложенных папок будет отображен на листе в виде подробной таблицы:

	A	B	C	D	E
1	Имя файла	Путь	Размер	Дата создания	Дата изменени
2	ACDC - 01 - T.N.T..mp3	D:\Music\ACDC\ACDC - 01 - T.N.T..mp3	5158171	06.11.2011 10:24	10.12.2011 10:
3	ACDC - 02 - Problem Child.mp3	D:\Music\ACDC\ACDC - 02 - Problem Child.mp3	7800091	06.11.2011 10:24	24.11.2011 9:
4	ACDC - 03 - Whole Lotta Rosie.mp3	D:\Music\ACDC\ACDC - 03 - Whole Lotta Rosie.mp3	8018266	06.11.2011 10:24	24.11.2011 9:
5	ACDC - 04 - Sin City.mp3	D:\Music\ACDC\ACDC - 04 - Sin City.mp3	6848398	06.11.2011 10:24	24.11.2011 9:
6	ACDC - 05 - Highway To Hell.mp3	D:\Music\ACDC\ACDC - 05 - Highway To Hell.mp3	5006233	06.11.2011 10:24	10.12.2011 10:
7	ACDC - 06 - Girl's Got Rhythm.mp3	D:\Music\ACDC\ACDC - 06 - Girl's Got Rhythm.mp3	4883353	06.11.2011 10:24	10.12.2011 10:
8	ACDC - 07 - If You Want Blood (You've Got It).mp3	D:\Music\ACDC\ACDC - 07 - If You Want Blood (You've Got It).mp3	6640662	06.11.2011 10:24	10.12.2011 10:
9	ACDC - 08 - Dirty Deeds Done Dirt Cheap.mp3	D:\Music\ACDC\ACDC - 08 - Dirty Deeds Done Dirt Cheap.mp3	5545272	06.11.2011 10:24	13.11.2007 20:
10	ACDC - 09 - Big Balls.mp3	D:\Music\ACDC\ACDC - 09 - Big Balls.mp3	3790470	06.11.2011 10:24	13.11.2007 20:
11	ACDC - 10 - Ride On.mp3	D:\Music\ACDC\ACDC - 10 - Ride On.mp3	8448626	06.11.2011 10:24	13.11.2007 20:
12	ACDC - 11 - Hells Bells.mp3	D:\Music\ACDC\ACDC - 11 - Hells Bells.mp3	7500822	06.11.2011 10:24	10.12.2011 10:
13	ACDC - 12 - Back In Black.mp3	D:\Music\ACDC\ACDC - 12 - Back In Black.mp3	6130334	06.11.2011 10:24	10.12.2011 10:

Создание резервных копий ценных файлов

При работе с некоторыми особо ценными файлами лучше бы периодически в течение рабочего дня сохранять их промежуточные версии, чтобы иметь возможность к ним вернуться. Выполнять постоянно вручную команды **Файл – Сохранить как** утомительно, да и случайно можно вместо **Сохранить как** нажать на **Сохранить**, похоронив этим большой кусок работы.

Если у вас Excel 2010 или новее, то кроме стандартного автосохранения у вас должна работать система версий – каждый раз при автосохранении Excel делает отдельную копию вашего текущего файла и (даже!) при выходе из программы и отрицательном ответе на вопрос "Сохранить изменения в файле?" все равно сохраняет временную копию. Добраться до этих временных копий можно через вкладку **Файл – Сведения – Версии (File – Properties – Versions)**:



Альтернативой может стать небольшой макрос, который сохраняет текущую книгу в заданную папку, добавляя к имени книги текущую дату и время в формате ДД-ММ-ГГ ЧЧ-ММ (например, *Мой проект 12-10-12 07-35.xlsx*). Периодически запуская этот макрос на ключевых этапах работы с файлом, я получаю список из энного количества файлов-версий рабочей книги и, соответственно, легко могу откатиться к нужному варианту в прошлом.

Откройте редактор Visual Basic, выбрав на вкладке **Разработчик – Редактор Visual Basic (Developer – Visual Basic Editor)** или нажав **Alt+F11**. Вставьте через меню **Insert – Module** новый пустой модуль и введите туда текст этого макроса:

```
Sub Backup_Active_Workbook()
    Dim x As String
    strPath = "c:\TEMP" 'папка для сохранения резервной копии
    On Error Resume Next
    x = GetAttr(strPath) And 0
    If Err = 0 Then ' если путь существует – сохраняем копию книги, добавляя дату-время
        strDate = Format(Now, "dd-mm-yy hh-mm")
        FileNameXls = strPath & "\" & "Мой проект" & " " & strDate & ".xls" 'или xlsm
        ActiveWorkbook.SaveCopyAs Filename:=FileNameXls
    Else 'если путь не существует – выводим сообщение
        MsgBox "Папка " & strPath & " недоступна или не существует!", vbCritical
    End If
End Sub
```

Естественно, путь к папке (C:\TEMP) и имя файла (*Мой проект*) надо заменить на свои.

Если ваша папка для сохранения находится на сетевом диске, то ее адрес можно прописать, используя IP-адрес сервера, например:

```
strPath = "\\192.168.1.1\Папка для бэкапов"
```

Еще одно, возможно, полезное дополнение в том, что имя файла может быть не постоянным, а браться из заданной ячейки листа, где его либо вводит пользователь, либо оно автоматически формируется формулами (например, функцией СЦЕПИТЬ и т.д.). Тогда необходимо будет чуть подправить следующую строку:

```
FileNameXls = strPath & "\" & Sheets("Лист1").Range("A1").Value & " " & strDate & ".xlsx"
```

Предполагается, что имя файла берется с листа *Лист1* из ячейки A1.

Также полезным в некоторых случаях может быть включение этого макроса в автозагрузку, чтобы бэкап автоматически выполнялся при открытии и закрытии книги. Для этого в редакторе Visual Basic откройте в левом верхнем углу двойным щелчком модуль **ЭтаКнига (ThisWorkbook)** и впишите туда команды для вызова нашего макроса:

