

УДК 004.4
ББК 32.973.26-018.2
К63

Комолова, Н. В.

К63 Программирование на VBA в Excel 2019. Самоучитель / Н. В. Комолова, А. В. Клименко. — СПб.: БХВ-Петербург, 2020. — 496 с.: ил. — (Самоучитель)

ISBN 978-5-9775-6593-6

Книга научит самостоятельно создавать приложения для автоматизации работы в программе Microsoft Office Excel 2019 с использованием макросов и языка программирования Visual Basic for Applications (VBA). Приведена информация о новинках программы, а также сервисах Power по работе с данными. Даны теоретические сведения о программировании, элементах объектной модели Excel, запуске и отладке макросов. Рассмотрены вопросы автоматизации рабочего листа при помощи элементов управления Excel. Описаны приемы создания макросов, пользовательских функций и форм в редакторе VBE. Приведены способы взаимодействия при помощи VBA с другими программами пакета Microsoft Office. Для закрепления материала рассмотрены примеры пользовательских приложений с анализом и поясняющими комментариями. Основные термины VBA и редактора VBE приведены в глоссарии. Файлы рабочих книг с поддержкой макросов для каждой главы размещены на сайте издательства.

Для широкого круга пользователей

УДК 004.4
ББК 32.973.26-018.2

Группа подготовки издания:

Руководитель проекта	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Екатерина Сависте</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Дизайн серии	<i>Марины Дамбиевой</i>
Оформление обложки	<i>Карины Соловьевой</i>

"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

ISBN 978-5-9775-6593-6

© ООО "БХВ", 2020
© Оформление. ООО "БХВ-Петербург", 2020

Оглавление

Введение	11
Язык VBA	11
Новинки Excel 2019 и службы Power	11
Зачем нужен VBA в Excel 2019?	17
Объектно-ориентированное программирование	18
Объект	19
Объектная модель Microsoft Excel 2019	19
Объектная модель VBA	20
Объект <i>Application</i>	20
Классы	21
Свойства	22
Методы	22
События	24
Глава 1. Основные понятия VBA	25
Базовые конструкции языка Visual Basic	25
Типы данных в VBA	26
Константы и переменные, переменные объектов	28
Область действия переменных и констант	28
Объявление констант и переменных	29
Начало работы	29
Настройка безопасности	32
Запись макроса	34
Имя макроса	34
Разработка проекта	35
Создание модуля	37
Создание модуля с помощью контекстно-зависимого меню	38
Окно кода	38
Оператор <i>Option Explicit</i>	39
Первая процедура	40
Объявление переменной в VBA	41
Оператор <i>Debug.Print</i>	42
Автоматический ввод атрибутов команд	44

Структура кода процедуры	45
Метод <i>Worksheets.Activate</i>	45
Активная ячейка <i>ActiveCell</i>	46
Открытие книги с макросом	47
Ввод данных	49
Оператор <i>With</i>	50
Свойство <i>Selection</i>	51
Свойство <i>Orientation</i>	52
Объект <i>Range</i>	53
Кнопка (элемент управления ActiveX)	54
Свойство <i>Offset</i>	55
Функция <i>Environ</i>	56
Функция <i>MsgBox</i>	57
Константы табуляции <i>Chr(9)</i> и перевода строки <i>Chr(10)</i>	59
Диалоговое окно, создаваемое функцией <i>InputBox</i>	60
Переменная типа <i>String</i>	61
Переменная типа <i>Long</i>	61
Переменная типа <i>Byte</i>	62
Методы <i>Protect</i> и <i>Unprotect</i>	62
Запуск макроса при помощи нажатия сочетания клавиш	64
Как удалить модуль?	66
Глава 2. Основы программирования в VBA	69
Ячейка и диапазон ячеек	69
Арифметические выражения	70
Правила записи арифметических выражений	71
Арифметические выражения в ячейке	71
Арифметические выражения с ячейками	72
Обрамление ячейки — метод <i>BorderAround</i>	73
Оператор <i>With</i>	74
Вложенные операторы <i>With</i>	75
Генерация случайных чисел <i>RAND</i>	76
Перевод градусов по Фаренгейту в градусы по Цельсию	77
Замена значений формул числом	78
Работа с цветом	81
Функция <i>RGB</i>	81
Свойства <i>Color</i> и <i>ColorIndex</i>	82
Свойство <i>Color</i>	82
Свойство <i>ColorIndex</i>	82
Палитра цветов	86
Подсчет цветов в рисунке	88
Заливка ячейки цветом	91
Защита ячеек рабочего листа от форматирования	93
Выделение ячеек по цветовому соответствию в диапазоне	94
Заливка строк с заданным шагом	95
Выделение миганием	96
Календарь	98
Заливка ячеек, содержащих формулы	100
Подсветка минимального и максимального значений	101
Цветовая шкала	103

Глава 3. Логические операторы	105
Оператор <i>If...Then...Else</i>	105
Операторы сравнения.....	105
Неполная форма оператора <i>If...Then</i>	107
Полная форма оператора <i>If...Then...Else</i>	108
Оператор <i>ElseIf</i>	108
Вложенные логические операторы	109
Примеры использования логических операторов.....	110
Свойство <i>Name</i>	110
Свойство <i>Value</i>	110
Функции <i>IsEmpty</i> и <i>IsNumeric</i>	111
Свойство <i>Range.HasFormula</i>	111
Переход к ячейке A2019.....	112
Пример с оператором <i>Case</i>	113
Функция <i>InputBox</i>	115
Оператор <i>GoTo</i>	116
Проверка существования файла	117
Свойства объекта <i>Application</i>	118
Глава 4. Операторы цикла	119
Цикл <i>For...To...Step...Next</i>	119
Цикл <i>For...To...Next</i>	120
Заполнение столбца	121
Заполнение столбца с большим шагом	122
Отрицательный шаг	122
Выход из цикла по условию.....	123
Кнопка для запуска макроса (элемент управления формы).....	124
Вложенный цикл <i>For...To...Next</i>	125
Цикл <i>For...Each</i>	127
Цикл <i>Do...Loop</i>	128
Цикл <i>While...Wend</i>	131
Время работы программы.....	132
Глава 5. Функции, определенные пользователем	139
Построение функций	139
График функции одной переменной	139
Структура кода функции пользователя.....	140
График функции одной переменной (продолжение)	140
График кусочно-непрерывной функции с двумя условиями	143
График кусочно-непрерывной функции с тремя условиями.....	145
Названия формул на английском языке.....	148
Пользовательская функция с тремя аргументами.....	149
Создание собственной категории.....	150
Функция без параметров	152
Переименование рабочего листа	152
Функция с аргументом типа <i>Range</i>	153
Функция с массивом	155
Функция с массивом в качестве аргумента	156
Вызов функции из процедуры	158

Вызов процедуры, использующей функцию, из другой процедуры	159
Запись названий формул	161
Вычисление определенного интеграла	162
Метод прямоугольников	163
Метод трапеций	163
Метод Симпсона	164
Переключатели <i>OptionButton</i>	167
Глава 6. Пользовательская форма.....	171
Создание форм средствами VBA	171
Форма <i>UserForm</i>	171
Семейство форм	172
Свойства формы.....	172
Разметочная сетка	174
Методы формы.....	175
События формы	175
Командная кнопка для показа формы	176
Элементы управления	178
Префиксы	181
Элемент управления <i>Label</i>	181
Элемент управления <i>CommandButton</i>	184
Элемент управления <i>TextBox</i>	187
Элементы управления <i>OptionButton</i> и <i>Frame</i>	190
Ключевое слово <i>Me</i>	193
Элемент управления <i>ScrollBar</i>	194
Элемент управления <i>ListBox</i>	198
Элемент управления <i>ComboBox</i>	201
Элемент управления <i>Image</i>	204
Элемент управления <i>SpinButton</i>	209
Элемент управления <i>TabStrip</i>	213
Элементы управления <i>CheckBox</i> и <i>MultiPage</i>	216
Элемент управления <i>RefEdit</i>	220
Элемент управления <i>ToggleButton</i>	223
Пользовательский элемент управления	225
Элементы управления формы.....	226
Элемент управления <i>Полоса прокрутки</i>	227
Глава 7. Программирование объектов <i>Shape</i>	231
Типы объектов, свойства и методы семейства <i>Shapes</i>	231
Тип объекта <i>msoShapeRectangle</i> (прямоугольник) с заливкой (<i>Fill</i>)	232
Тип объекта <i>msoConnectorCurve</i> (соединительная линия).....	234
Метод <i>AddConnector</i>	235
Метод <i>Patterned</i>	236
Рисование линии: метод <i>AddLine</i>	238
Тип объекта <i>msoShapeSmileyFace</i>	239
Свойство <i>Name</i>	240
Стрелка	241
Метод <i>FillFormat.OneColorGradient</i>	243
Текстовый фрейм.....	243
Тип фигуры <i>msoShapeHeart</i> (сердце) с заливкой (<i>Fill</i>).....	245

Метод <i>Group</i>	246
Создание выноски с текстовым фреймом.....	247
Свойство <i>ThreeD</i>	248
Частичное и полное удаление фигур.....	250
Оператор <i>Set</i>	252
Создание собственных элементов инфографики.....	253
Фракталы.....	255
Тип данных, определенный пользователем.....	256
Фракталы из треугольников.....	257
Фракталы из многоугольников.....	261
Фракталы из четырехугольников.....	263
Глава 8. Работа с ячейками и областями.....	267
Объект <i>Application</i>	267
Свойства объекта <i>Application</i>	267
Методы объекта <i>Application</i>	268
Объект <i>Range</i>	269
Адресация ячеек.....	269
Свойства объекта <i>Range</i>	270
Методы объекта <i>Range</i>	271
Объект <i>Selection</i>	272
Объект <i>Cell</i>	272
Выделение нескольких областей.....	272
Выделение последней ячейки в диапазоне.....	273
Свойство <i>Range.End</i>	275
Выделение ячеек с формулами.....	276
Выделение используемого диапазона данных.....	278
Форматирование объединенных ячеек.....	278
Выделение по условию.....	279
Удаление символов из ячеек.....	280
Убираем текст.....	282
Имена и фамилии.....	283
Метод <i>Delete</i>	284
Метод <i>Clear</i>	285
Метод <i>Application.Goto</i>	286
Скрытие данных.....	286
Копирование и специальная вставка.....	286
Поиск минимума и максимума в диапазоне.....	287
Глава 9. Работа с данными.....	289
Массив из трех элементов.....	289
Динамический массив данных.....	291
Сравнение областей на одном листе.....	293
Сравнение областей на разных листах.....	294
Сортировка.....	298
Сортировка диапазона данных.....	300
Сортировка областей (блоков).....	301
Простая сортировка блоков.....	301
Сортировка блоков с изменением ее условий.....	303
Сортировка по цвету.....	305

Контроль автофильтра посредством VBA.....	308
Команда <i>Итоги</i>	308
Сортировка данных при помощи <i>Среза</i>	311
Сводные таблицы <i>PivotTable</i>	316
Глава 10. Автоматизация диаграмм.....	319
Объектная модель диаграмм.....	319
Типы диаграмм.....	321
Свойства объекта <i>Chart</i>	324
Методы объекта <i>Chart</i>	324
Первая диаграмма.....	325
Создание диаграммы с помощью VBA.....	328
Коническая гистограмма.....	332
Печать диаграмм.....	334
Удаление диаграммы.....	336
Форматирование параметров диаграммы.....	337
Форматирование цветов поверхности.....	339
Добавление линии тренда.....	341
Геолокация.....	345
Изменение прозрачности.....	347
Красивые узоры.....	349
Глава 11. Программирование объектов и событий.....	353
Где и как создаются процедуры обработки событий?.....	354
Процедура для объекта <i>ЭтаКнига</i>	355
События, связанные с нажатием кнопок мыши.....	356
Процедура в модуле.....	356
Событие для объекта <i>Worksheet</i> (Лист).....	357
Ключевое слово <i>ByVal</i> и параметр <i>Target</i>	358
Очистка ячейки.....	359
Свойства <i>ScrollRow</i> и <i>ScrollColumn</i>	360
События активации и деактивации.....	360
Свойство приложения <i>ActiveWindow</i>	360
Активный лист.....	361
Число обращений к макросу.....	362
Управление выделением области.....	363
События <i>Activate</i> и <i>Deactivate</i> рабочего листа.....	364
Двойной щелчок левой кнопкой мыши.....	365
Щелчок правой кнопкой мыши.....	366
Введите пароль.....	366
Событие закрытия книги.....	368
Событие сохранения книги.....	368
Глава 12. Операторы даты и времени.....	371
Вывод даты и времени в окно <i>Immediate</i> оператором <i>Debug.Print</i>	371
Печать даты и времени с помощью функции <i>CDate</i>	372
Функции <i>DateSerial</i> и <i>TimeSerial</i>	373
Текущие дата и время.....	373
Текущие дата и время с учетом минут и секунд.....	374

Функция <i>Weekday</i> — день недели	374
Функция <i>Format</i>	375
Функция <i>DateDiff</i>	377
Функция <i>DatePart</i>	377
Функция <i>WeekdayName</i>	378
Вывод сообщения на 3 секунды	379
Метод <i>Application.OnTime</i>	380
Автоматическое заполнение ячеек датами методом <i>AutoFill</i>	381
Подсветка даты	383
Поиск даты	385
Календарь	386
Календарь по месяцам	389
Календарь по неделям	392
Определение возраста	395
Глава 13. Действия с рабочей книгой	397
Свойства объекта <i>Workbook</i>	397
Методы объекта <i>Workbook</i>	398
Событие и метод <i>Open</i>	399
Открытие рабочей книги методом <i>Workbooks.Open</i>	400
Свойство <i>Application.Dialogs</i> для работы со встроенными диалоговыми окнами	401
Открытие рабочей книги в диалоговом окне	402
Открытие приложения Блокнот	404
Свойство <i>Workbook.Name</i>	405
Создание рабочей книги	406
Имя приложения	406
Сохранение рабочей книги	408
Метод <i>Workbook.Save</i>	408
Метод <i>Workbook.SaveAs</i>	408
Метод <i>Workbook.SaveCopyAs</i>	411
Сохранение всех книг и выход из программы	412
Сохранение всех книг и выход по запросу	413
Защита рабочей книги методом <i>Workbook.Protect</i>	413
Объект <i>Worksheet</i>	414
Свойства объекта <i>Worksheet</i>	414
Методы объекта <i>Worksheet</i> и семейства <i>Worksheets</i>	414
Защита рабочего листа методом <i>Worksheet.Protect</i>	415
Деление рабочего листа на страницы для печати	419
Глава 14. Файловые операции	421
Форматы файлов Microsoft Excel	421
Метод <i>CreateTextFile</i> для объекта <i>FileSystemObject</i>	422
Список файлов указанной папки	423
Режим доступа <i>Input/Output</i>	424
Файлы из <i>Application.AddIns</i>	426
Объект <i>FileDialog</i>	428
Функция <i>GetAttr</i>	430
Документирование информации о файле	431
Проверка существования файла	433

Оператор <i>Kill</i> для удаления файла.....	434
Оператор <i>FileCopy</i> для копирования файла	435
Переименование файла	436
Перемещение файла	436
Глава 15. Отладка программ и сообщения об ошибках	439
Возникновение ошибок.....	439
Выявление и исправление ошибок.....	440
Три окна для просмотра ошибок.....	442
Окно просмотра <i>Immediate</i>	442
Окно <i>Locals</i>	444
Окно наблюдения <i>Watches</i>	445
Объект <i>Err</i>	448
Оператор <i>On Error</i>	448
Оператор <i>On Error Resume Next</i>	449
Оператор <i>On Error GoTo</i> : вариант 1.....	450
Оператор <i>On Error GoTo</i> : вариант 2.....	450
Константы <i>xlDisabled</i> и <i>xlInterrupt</i> свойства <i>Application.EnableCancelKey</i>	452
Массив листов.....	453
Команда меню <i>Debug</i>	453
Глава 16. Программирование связей	455
Гиперссылки.....	455
Кнопка гиперссылки.....	457
Передача данных из Excel в Word.....	458
Внедрение документа Word в Excel	459
Передача данных из Excel в PowerPoint	461
Передача данных из PowerPoint в Excel	463
Приложение 1. Глоссарий терминов Visual Basic for Applications	465
Приложение 2. Глоссарий терминов Visual Basic Editor	469
Приложение 3. Описание электронного архива.....	485
Предметный указатель	487

Введение

Программа Microsoft Excel, наряду с Microsoft Word, Microsoft Access, Microsoft PowerPoint, представляющая собой приложение программного пакета Microsoft Office, весьма популярна и технически хорошо разработана. Но даже если вы с ней ежедневно работали, то могли не заметить еще одно ее удивительное достоинство — возможность программирования на языке *Visual Basic for Applications* (VBA).

Как известно, VBA — встроенный язык программирования всех приложений Microsoft Office. Соответственно, на VBA можно программировать и в Microsoft Word, и в Microsoft Access, и в других приложениях этого пакета.

Центральное понятие VBA — *макрос*, или *макрокоманда* — последовательность команд на языке VBA, сохраненных под каким-либо именем.

Язык VBA

Язык VBA освоить не трудно, это достаточно легкий язык программирования, и с его помощью можно быстро получить ощутимые результаты — профессионально сконструированные приложения, решающие практически все задачи в операционной среде Windows.

Visual Basic for Applications относится к числу объектно-ориентированных языков, т. е. при разработке проектов на VBA все данные и действия над ними представляются в виде объектов.

VBA основан на технологии визуального программирования, предлагающей конструирование рабочей поверхности приложения и элементов его управления непосредственно на экране, а также запись всей программы или ее частей при помощи макрорекордера.

Новинки Excel 2019 и службы Power

Microsoft Excel в новой версии конечно поражает своим многообразием инструментов и имеющихся дополнительных служб. Все функции мы, разумеется, рассматривать не будем, остановимся на основных новинках Microsoft Excel 2019.

Итак, появились новые типы диаграмм **Воронка**, **Карта** (для визуализации геолокационных данных). Облегчает работу расширенная подсказка по выбору функции при вводе названия функции в строке формул (рис. В1).

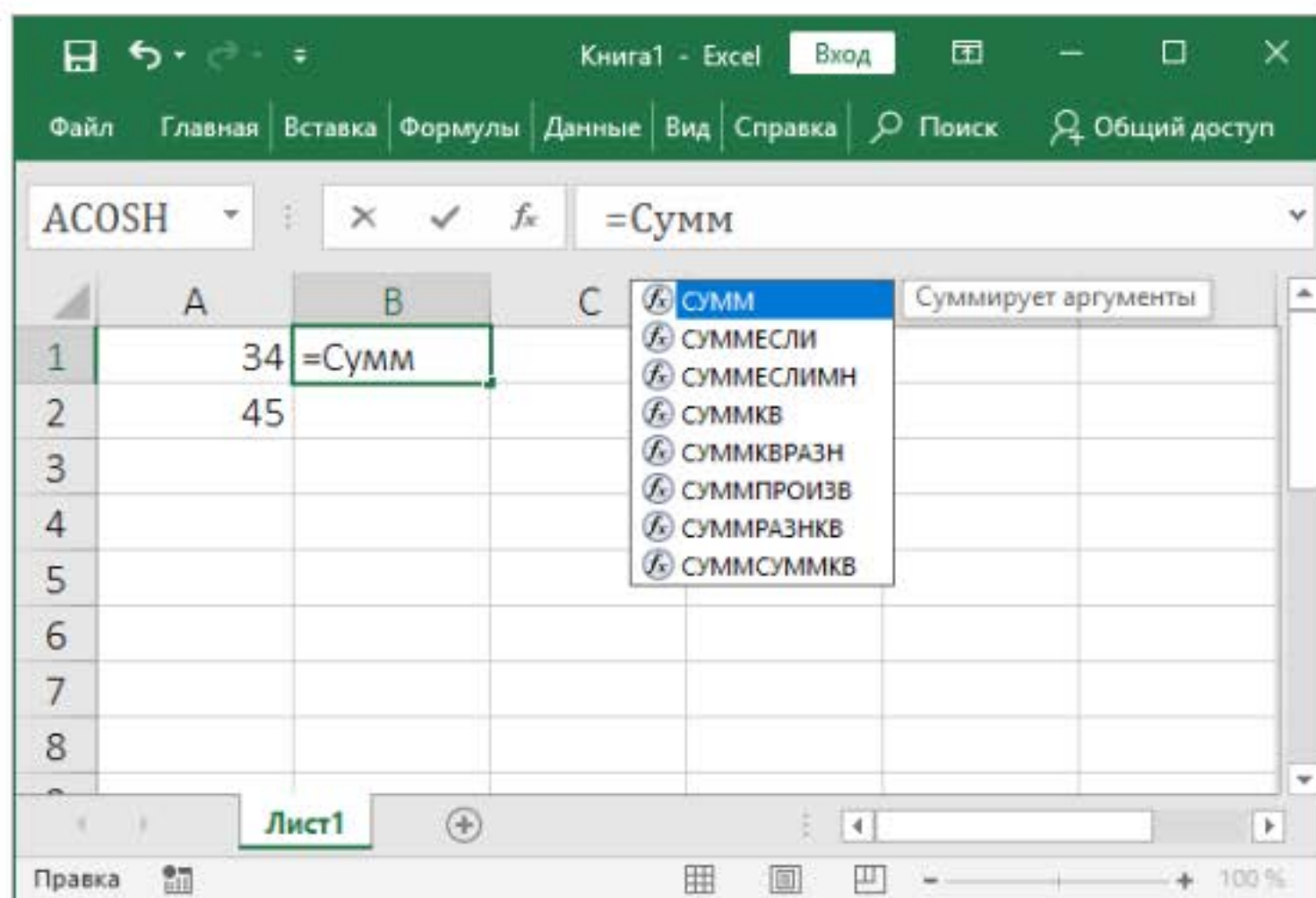


Рис. В1. Всплывающее меню с возможностью выбора функции

В программе реализованы и улучшены очень мощные сервисы по работе с данными, такими как Power Query, Power Pivot, Power View и Power Map.

Power Query — это технология подключения к данным, с помощью которой можно обнаруживать, подключать, объединять и уточнять данные из различных источников для последующего анализа. Доступна загрузка данных из файлов различных форматов, из целого множества баз данных (рис. В2), в том числе созданных с использованием профессиональных СУБД, из веб-служб и других источников. Для импорта данных табличного типа устанавливается соединение с выбранной базой данных и через соответствующие диалоговые окна осуществляется импорт. При этом таблицы автоматически преобразуются в таблицы с управляющими элементами для фильтрации столбцов. Настройка Power Query ранее находилась на отдельной вкладке Excel, теперь доступ к ней осуществляется при помощи перехода на вкладку **Данные** и использования команд в области **Получить и преобразовать данные**.

При успешном размещении таблиц на рабочих листах справа появляется окно **Запросы и подключения** с расширенной информацией о запросах и имеющихся подключениях к источникам информации (рис. В3). Переход в редактор Power Query осуществляется при двойном щелчке левой кнопкой мыши на строке с названием запроса (рис. В4).

Power Pivot — это технология моделирования данных, которая позволяет создавать модели данных, устанавливать отношения и добавлять вычисления. С помощью Power Pivot возможно работать с большими наборами данных, создавать разверну-

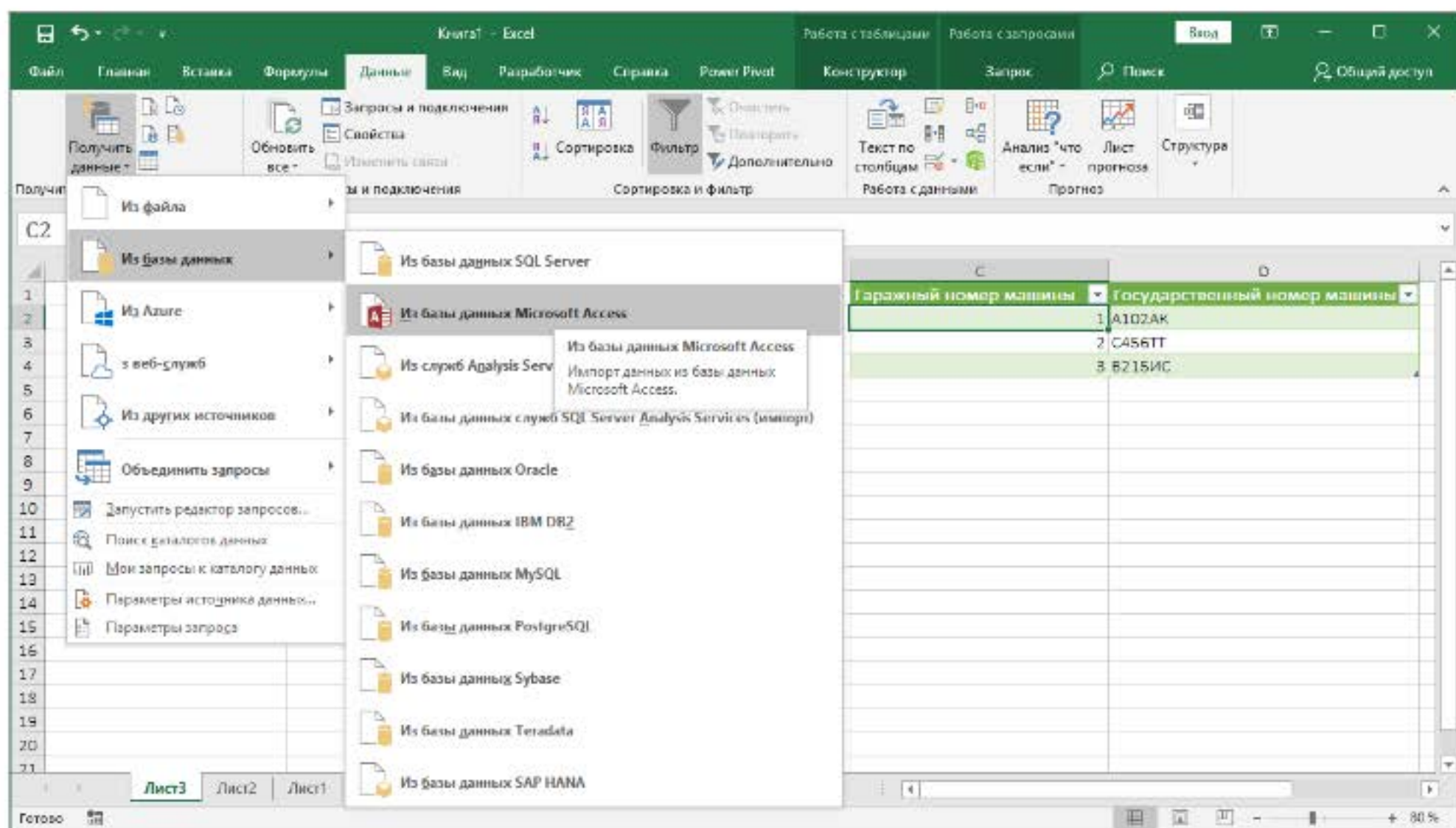


Рис. В2. Импорт таблицы из базы данных

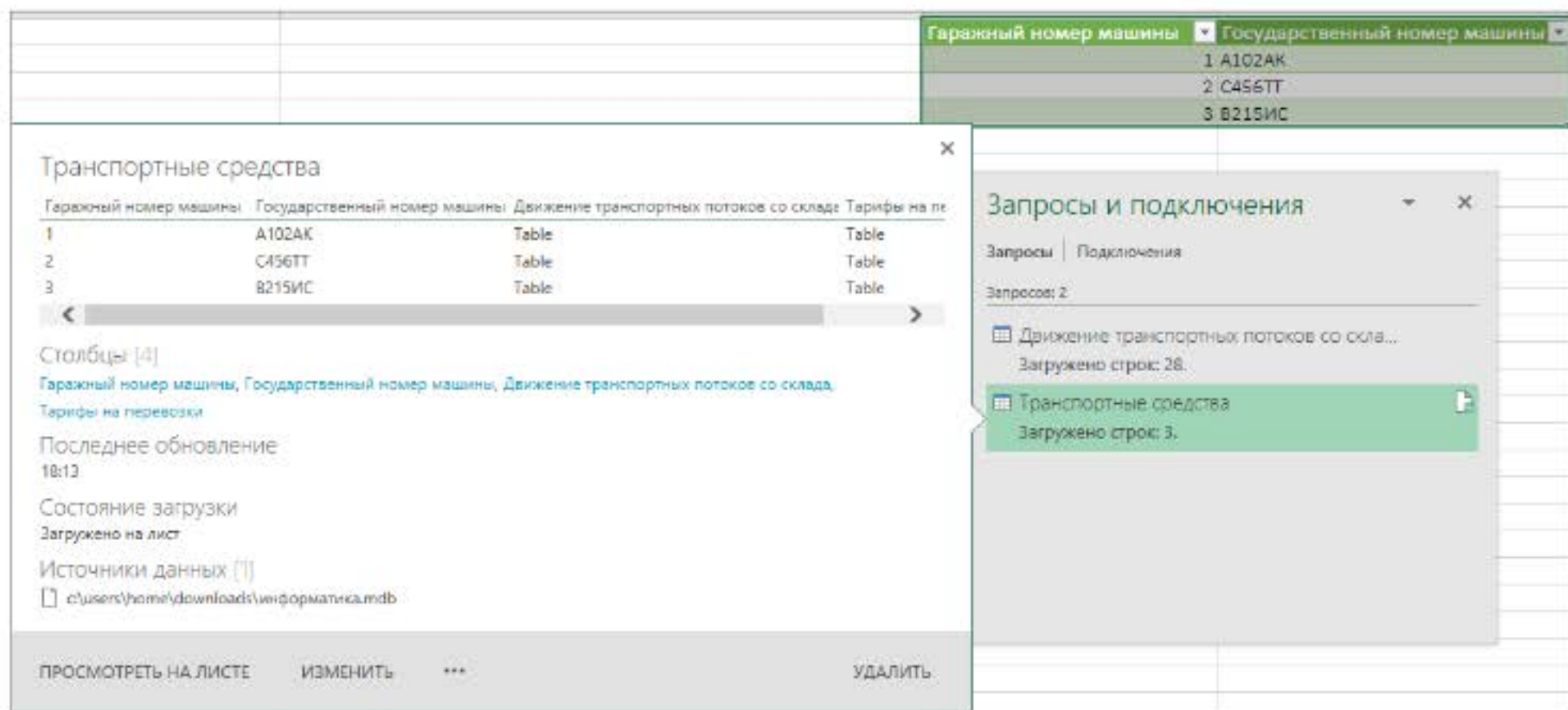


Рис. В3. Окно Запросы и подключения

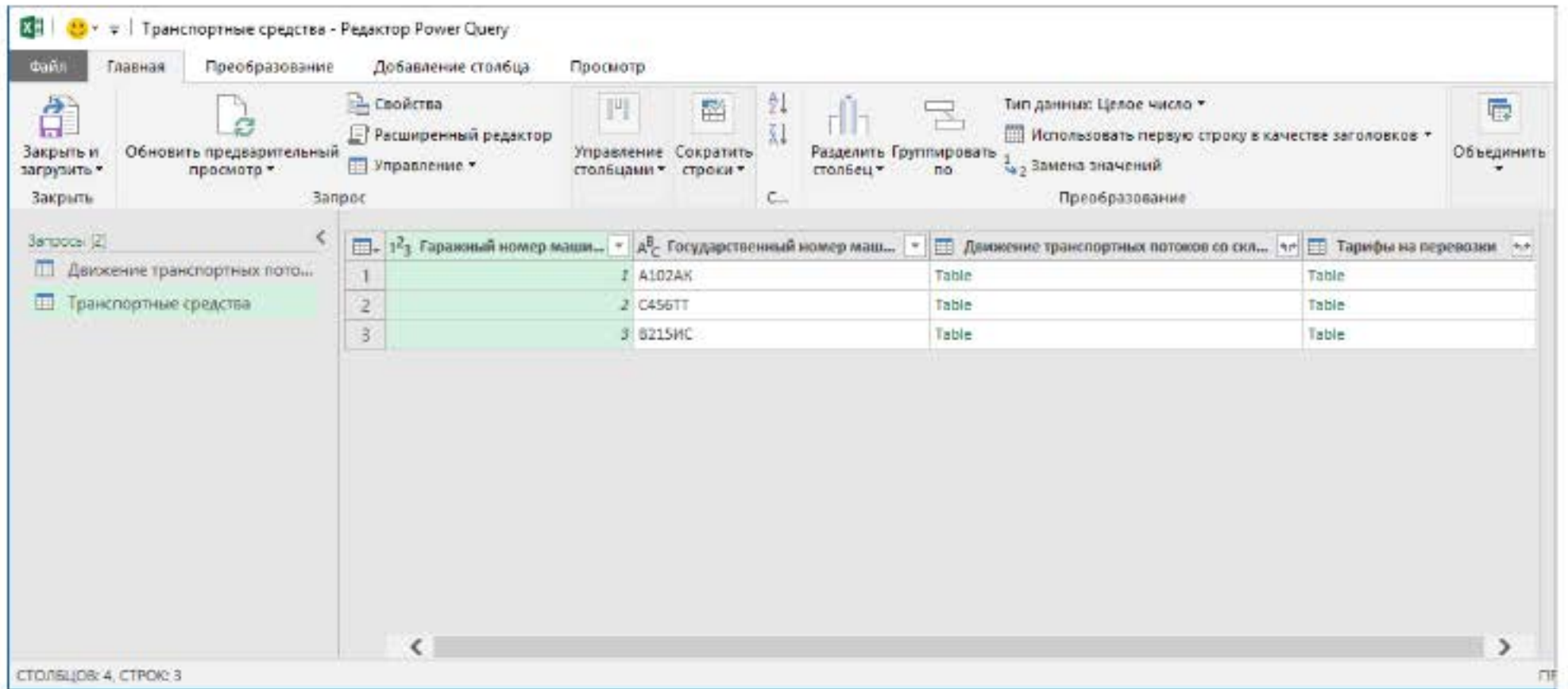


Рис. В4. Переход в редактор Power Query

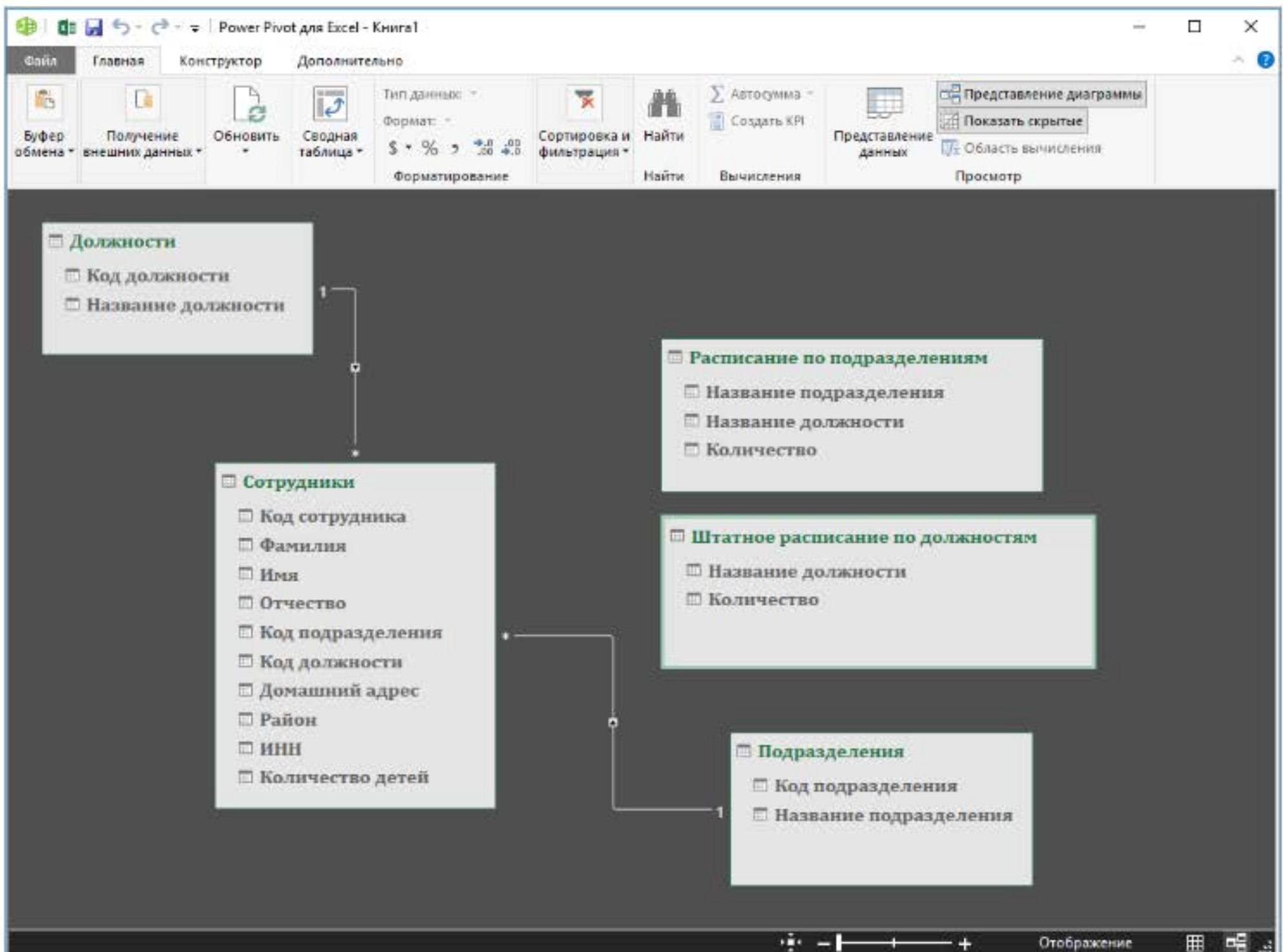


Рис. В5. Просмотр диаграммы — модели данных внедренной базы данных

тые отношения и сложные (или простые) вычисления. Работа с сервисом осуществляется при помощи вкладки **Power Pivot**. Также можно просмотреть модель импортированной базы данных в специальном окне **Power Pivot** (рис. В5).

Следующая надстройка, применяемая в Microsoft Excel для визуализации данных, с помощью которой можно создавать интерактивные диаграммы, графики, карты и другие наглядные элементы, позволяющие оживить информацию, — это технология *Power View*, доступная в Excel, SharePoint, SQL Server и Power BI. Управлять подключением надстройки можно при помощи диалогового окна **Параметры Excel**, вызываемого при помощи команды меню **Файл | Параметры**. Для подключения вкладки **Power View** перейдите в диалоговом окне в раздел **Настроить ленту** и поставьте галочку напротив **Power View** в списке вкладок в области **Основные вкладки** (рис. В6).

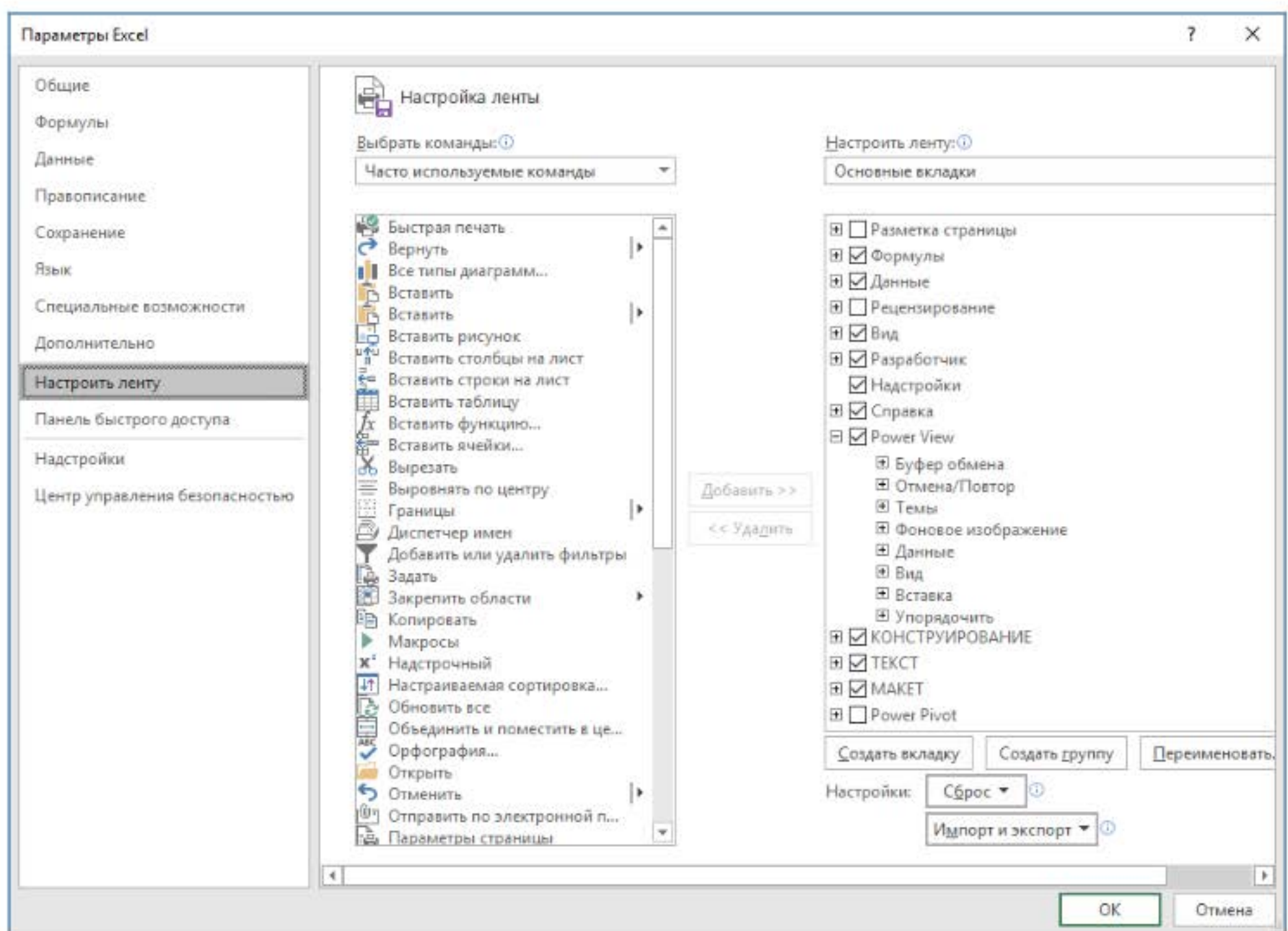


Рис. В6. Подключение надстройки Power View

Также в программе широко используется надстройка Power Map — надстройка, предназначенная для трехмерной геопространственной визуализации, — теперь интегрирована в Excel. Для вызова окна работы с надстройкой (рис. В7) выделите необходимый диапазон данных и на вкладке ленты **Вставка** в области **Обзоры** разверните список кнопки **3D-карта**. Укажите команду **Открыть в 3D Maps**.

На основе созданных визуализаций с использованием географических данных в Power Map можно создавать видеоролики (рис. В8) для наглядной иллюстрации

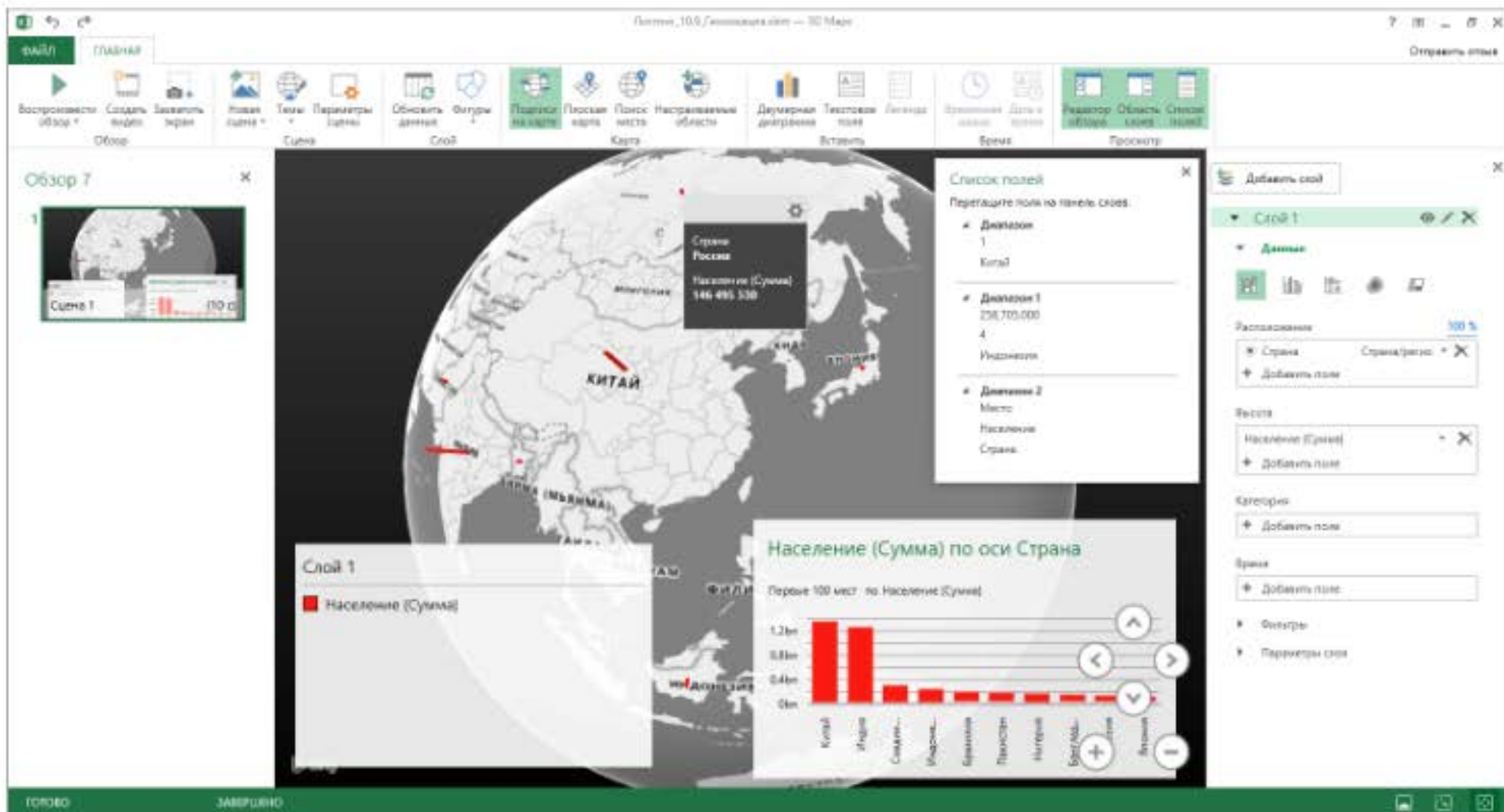


Рис. В7. Работа с надстройкой 3D Maps

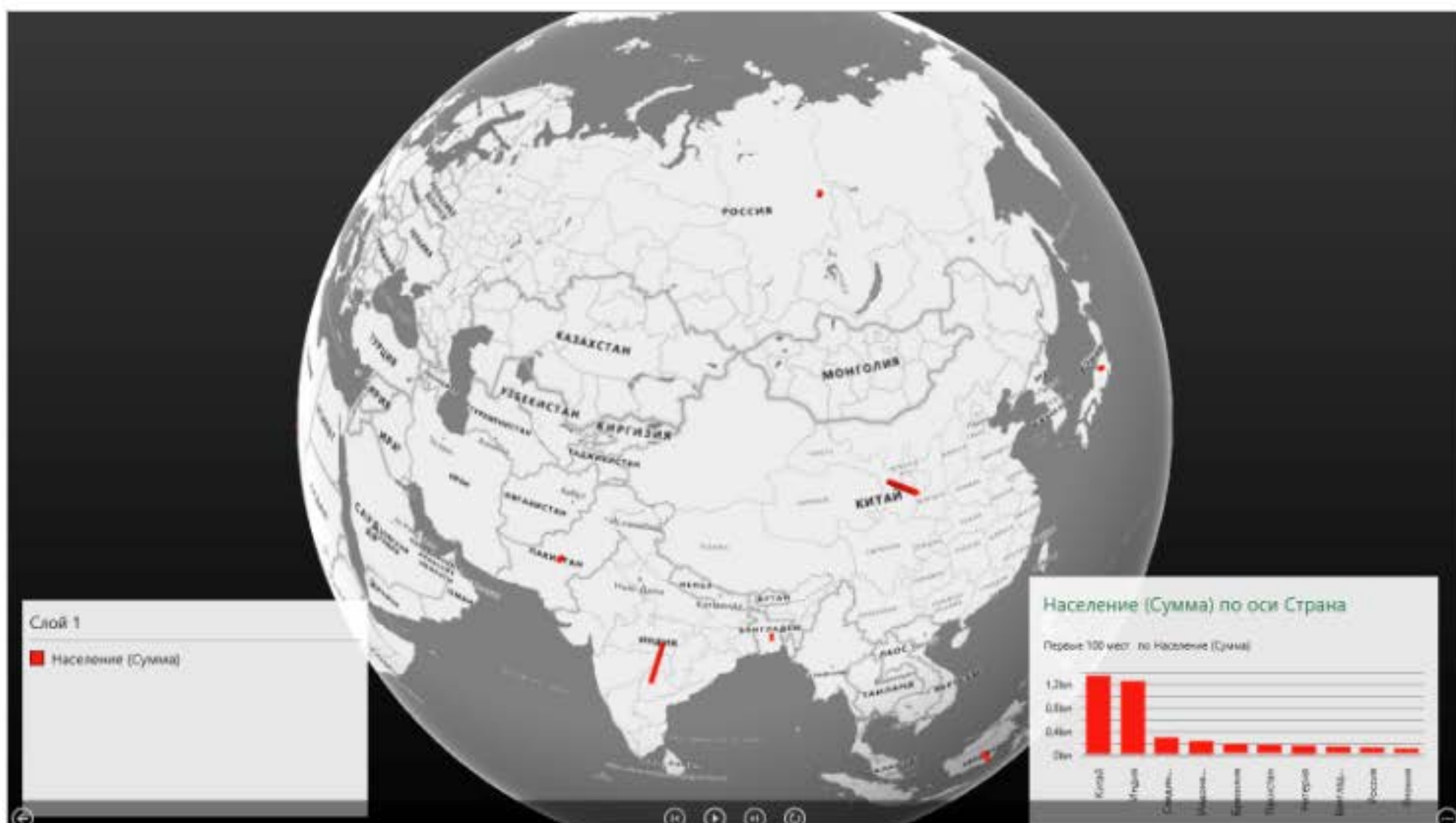


Рис. В8. Воспроизведение обзора в 3D Maps

данных таблицы на карте мира или вставки эффектного ролика в презентацию Power Point.

Среди новинок Microsoft Excel стоит отметить поддержку масштабируемой векторной графики в виде SVG-файлов, которые можно вставить на рабочий лист, пользовательскую форму и другие элементы и преобразовать в фигуры. Коллекция

значков, которые можно вставить для оформления таблиц и отчетов находится в диалоговом окне **Вставка значков**, вызываемом при помощи нажатия кнопки **Значки**, расположенной на вкладке ленты **Вставка** в группе **Иллюстрации**.

Зачем нужен VBA в Excel 2019?

С помощью макросов VBA можно автоматизировать ежедневную рутинную работу в программе Microsoft Excel — ведь действия, доступные пользователю на ее рабочем листе, составляют 10% от всех возможностей приложения.

Кроме того, VBA объединяет (интегрирует) приложения Microsoft Office, позволяет, не выходя из Excel, управлять работой других приложений и внедрять в Excel объекты из других приложений пакета.

Перечислим основные возможности VBA в Excel 2019.

- ◆ Автоматизированный доступ к рабочему листу, диапазону данных, ячейке.
- ◆ Защита VBA-кода, защита информации на уровне: рабочей книги, рабочего листа, ячейки с данными. Возможность установки защиты с различными параметрами.
- ◆ Работа с циклами и управление потоками данных.
- ◆ Работа с формулами с помощью стиля ссылок R1C1.
- ◆ Создание и управление имен в VBA.
- ◆ Доступ к встроенным функциям VBA и создание собственных функций.
- ◆ Обработка событий.
- ◆ Работа с массивами данных, в том числе с динамическими.
- ◆ Создание классов и коллекций.
- ◆ Автоматизация при помощи элементов управления формы и ActiveX.
- ◆ Создание пользовательских форм.
- ◆ Использование VBA для анализа данных и расширенной фильтрации.
- ◆ Работа со сводными таблицами при помощи VBA.
- ◆ Создание графиков и диаграмм.
- ◆ Визуализация данных, дополнительное форматирование и создание собственных элементов инфографики.
- ◆ Работа со спарклайнами.
- ◆ Считывание информации из Интернета и выкладывание информации в Интернет.
- ◆ Обработка текстовых файлов.
- ◆ Интеграция с другими приложениями Microsoft Office.
- ◆ Запросы данных из популярных баз данных.
- ◆ Многопользовательский доступ к данным.

- ◆ Работа с интерфейсом Windows API.
- ◆ Обработка ошибок различного рода.
- ◆ Создание надстроек, собственных команд меню, настройка ленты.
- ◆ Доступ к новым возможностям Microsoft Excel 2019.

Объектно-ориентированное программирование

Объектно-ориентированное программирование (ООП) вобрало в себя все лучшие идеи структурного программирования, объединив их с рядом новых понятий. В самом общем виде программа может быть организована одним из двух способов: *вокруг кода* (т. е. того, что фактически исполняется) или *вокруг данных* (т. е. того, что подвергается воздействию). Программы, созданные только методами структурного программирования, как правило, организованы вокруг кода. Такой подход можно рассматривать "как код, воздействующий на данные".

Объектно-ориентированные программы организованы вокруг данных, исходя из принципа: "данные управляют доступом к коду". В объектно-ориентированном языке программирования определяются данные и код, которому разрешается воздействовать на эти данные. Следовательно, тип данных точно определяет операции, которые могут быть выполнены над данными.

Все объектно-ориентированные языки программирования обладают тремя общими свойствами: инкапсуляцией, полиморфизмом и наследованием.

Инкапсуляция — это механизм программирования, объединяющий вместе код и данные, которыми он манипулирует, исключая как вмешательство извне, так и неправильное использование данных. В объектно-ориентированном языке данные и код могут быть объединены как бы в совершенно автономный "черный ящик". Внутри такого ящика находятся все необходимые данные и код. Когда код и данные связываются вместе подобным образом, создается объект. Иными словами, *объект* — это элемент, поддерживающий инкапсуляцию.

Полиморфизм — это свойство, позволяющее одному интерфейсу получить доступ к *общему классу действий*. Полиморфизм помогает упростить программу, позволяя использовать один и тот же интерфейс для описания общего класса действий. Выбрать конкретное действие (т. е. метод) в каждом отдельном случае — это задача компилятора.

Наследование представляет собой процесс, в ходе которого один объект приобретает свойства другого объекта. Это обеспечивает принцип *иерархической классификации*. Если пользоваться иерархиями, то для каждого объекта пришлось бы явно определять все его свойства. А если воспользоваться наследованием, то достаточно определить лишь те свойства объекта, которые делают его особенным в классе. Объект может также наследовать общие свойства своего родителя. Следовательно, благодаря механизму наследования один объект становится отдельным экземпляром более общего класса.

Таким образом, ООП — это совокупность принципов разработки программ, понятий и элементов языка, позволяющих успешно создавать программы большого объема. ООП базируется на связи в единое целое свойств и поведения предмета или процесса.

Свойства объекта — это данные, которые его характеризуют. Поведение задается функциями.

В программу Microsoft Excel 2019 встроены два языка программирования макросов: XLM и VBA. Изначально был создан язык XLM, но к настоящему моменту он устарел и полностью заменен на VBA. Тем не менее в Microsoft Excel 2019 можно запускать большинство XLM-макросов, а также создавать макросы этого типа. Но при этом невозможно записывать XLM-макросы. Поэтому для создания макросов следует использовать VBA.

Редактор VBA представляет собой окно, в котором осуществляется работа с кодом VBA. Это окно организовано на традиционном интерфейсе, предусматривающем использование меню и панелей инструментов.

Объект

Объект — это парадигма, основное понятие объектно-ориентированного программирования, означающее инкапсуляцию данных вместе с кодом, предназначенным для их обработки, т. е. объединение данных и кода в нечто целое, именуемое объектом.

Существует несколько определений объекта.

- ◆ Объект — это данные плюс функции.
- ◆ Объект — это инкапсулированная абстракция с четким протоколом доступа.
- ◆ Объект — это совокупность переменных состояния и связанных с ними методов.
- ◆ Объект — это сущность в адресном пространстве вычислительной системы, появляющаяся при создании экземпляров класса.
- ◆ Объект — это характеристика, назначенная элементу класса.

Объектная модель Microsoft Excel 2019

Объектная модель Microsoft Excel 2019 содержит свыше 280 объектов — полный их список приведен в файле Введение\Объектная модель Microsoft Excel 2019.doc сопровождающего книгу электронного архива.

ЭЛЕКТРОННЫЙ АРХИВ

Книгу сопровождает электронный архив, содержащий тексты всех сохраненных листингов, которые мы будем разрабатывать в процессе изучения ее материала (см. приложение 3).

Скачать электронный архив с FTP-сервера издательства можно по ссылке <ftp://ftp.bhv.ru/9785977565936.zip>, а также со страницы книги на сайте www.bhv.ru.

Объектная модель VBA

Объектная модель Microsoft Excel представляет собой иерархию объектов, подчиненных одному объекту — `Application` (само приложение Excel).

Объектов в VBA много (более 100), самые главные из них: `Application`, `Workbook` (Рабочая книга, т. е. ваш файл), `Worksheet` (Рабочий лист), `Worksheet Function` (Мастер функций).

Есть и другие, например `Range` (Диапазон), `Chart` (Диаграмма), `Cell` (Ячейка), `Row` (Строка), `Column` (Столбец), `Style` (Стиль), `Border` (Границы), `Interior` (Цвет фона), `Font` (Шрифт), `Shapes` (Фигуры), а также `Phonetic`, `Phonetics`, `TextEffectFormat`, `TickLabels`, `AxisTitle`, `ChartTitle`, `DataLabel`, `DataLabels`, `DisplayUnitLabel`, `Style`, `TextFrame`.

Многие объекты собраны в библиотеку объектов Microsoft Excel, но некоторые, например `Assistant`, входят в библиотеку Microsoft Office, общую для всех офисных приложений. На рис. В9 показаны основные компоненты объектной модели Microsoft Excel.

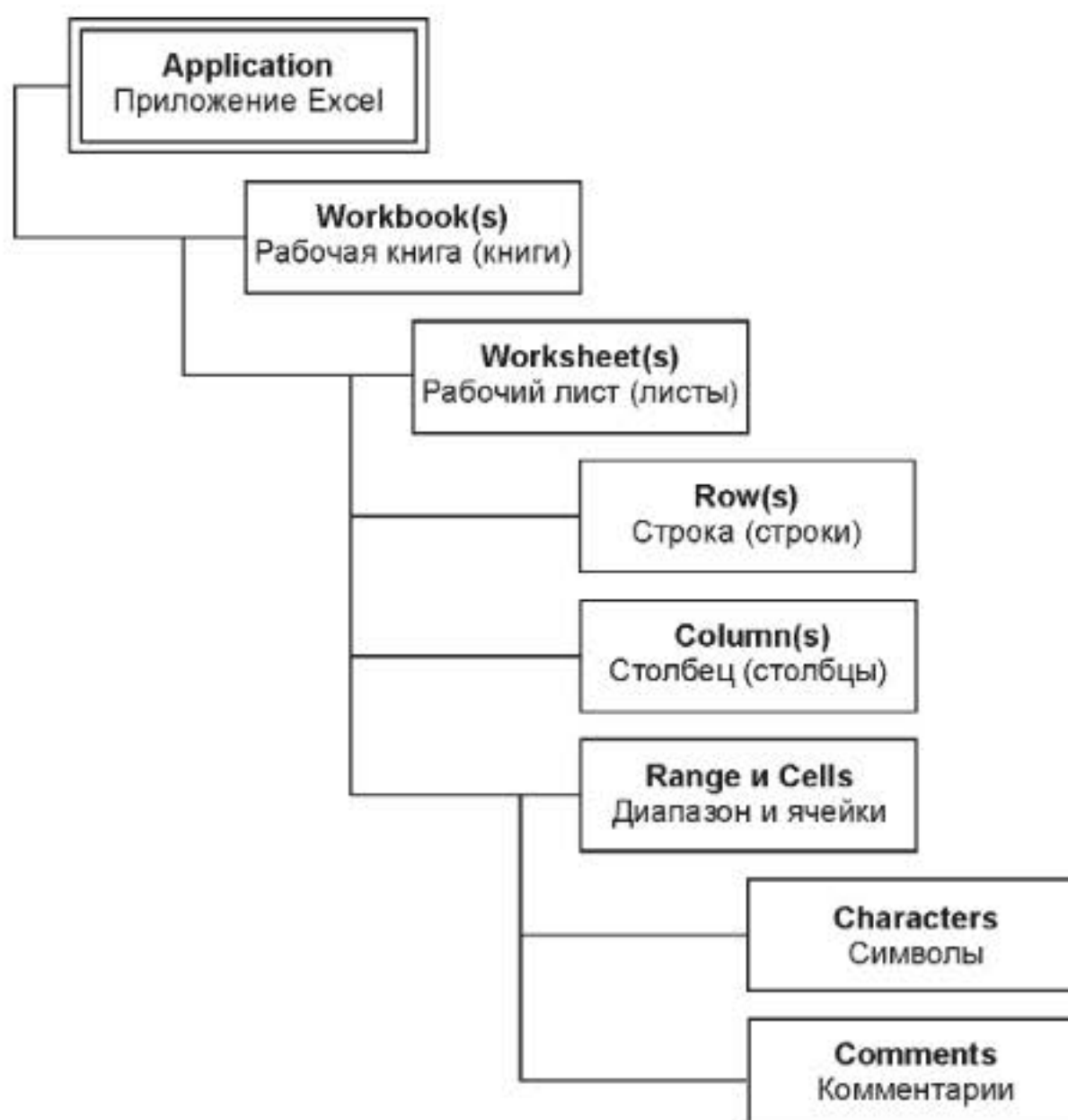


Рис. В9. Иерархия основных объектов объектной модели Microsoft Excel

Объект *Application*

Объект `Application` — это главный (корневой) объект Microsoft Excel, представляющий само приложение Microsoft Excel. Объект `Application` имеет очень много свойств, и чтобы понять, сколько потребуется времени, если писать программы со

всеми объектами, со всеми свойствами, методами, событиями, вы можете ознакомиться с полным их списком, содержащемся в файле Введение\Свойства объекта Application.doc, сопровождающего книгу электронного архива (см. приложение 3).

Точно так же с перечнем объектов объектной модели Microsoft Excel можно ознакомиться в файле Введение\Объектная модель Microsoft Excel 2019.doc, со списком методов объекта Application — в файле Введение\Методы объекта Application.doc, а со списком событий, управляющих объектом Application, — в файле Введение\События объекта Application.doc сопровождающего книгу электронного архива (см. приложение 3).

Итак, перечислим основные объекты VBA в порядке иерархии:

- ◆ Application (само приложение Excel);
 - Workbook (Рабочая книга — ваш файл);
 - Worksheet (Рабочий лист);
 - Range (Диапазон);
 - Chart (Диаграмма);
 - Character (Символ);
 - Style (Стиль);
 - Border (Граница);
 - Interior (Цвет фона);
 - Font (Шрифт);
 - WorksheetFunction (Мастер функций).

Множество некоторых объектов составляют семейства: Workbooks, Worksheets, Charts.

Классы

Класс — это одно из важнейших понятий ООП. Класс представляет собой разновидность абстрактного типа данных в ООП, характеризующийся способом своего построения. Класс описывает свойства и методы, которые будут доступны объекту, построенному по описанию, заложенному в класс.

Понятие "класс" подразумевает некоторое поведение и способ представления. Объект — это экземпляр класса.

Класс обычно описывается как проект, на основе которого впоследствии создается конкретный объект. Класс определяет имя объекта, его свойства и действия, над ним выполняемые.

В ООП классом называется определенный пользователем тип данных, который обладает внутренними данными и методами для работы с ними в форме процедур и функций.

Свойства

Объекты обладают *свойствами* (действиями над объектами).

Свойства характеризуют отличительные черты объектов, они сформированы для объектов. Например, свойство "Форматирование ячейки" следует искать для объекта "Ячейка".

Свойство (property) — это атрибут объекта, определяющий его характеристики, такие как надпись, размер, отступы, цвет заднего и переднего фона, положение на экране, гарнитура шрифта надписей, доступность, видимость и многие другие. При изменении значений свойств меняются характеристики объекта.

Синтаксис задания значений свойства:

Объект.Свойство = Значение_свойства

Приведем конкретный пример:

`Cells(3, 5).Value = 0`

Для объекта `Cells(3, 5)` (ячейка C5) устанавливается значение 0 при помощи свойства `Value`.

Методы

Объекты обладают *методами* (действиями самих объектов). Метод (method) — это действия, выполняемые над объектом, уведомления, получаемые или передаваемые объектом другим объектам или приложениям.

Когда речь идет о методе, то решается вопрос о выборе объекта. Основные методы объекта `Workbook` (Рабочая книга) приведены в табл. В1.

Таблица В1. Методы для рабочей книги Excel

Название метода	Назначение метода
Activate	Активировать первое окно, ассоциируемое с рабочей книгой
Close	Закреть книгу
PrintOut	Печатать книгу
Save	Сохранить изменения в текущем файле
SaveAs	Сохранить изменения книги в другом файле

В табл. В2 приведены основные методы для листа книги.

Таблица В2. Методы для листа рабочей книги Excel

Название метода	Назначение метода
Activate	Сделать текущий лист книги активным листом
Copy	Копировать рабочий лист

Таблица В2 (окончание)

Название метода	Назначение метода
Delete	Удалить рабочий лист
Move	Переместить рабочий лист
PrintOut	Напечатать рабочий лист
Protect/Unprotect	Защитить лист от изменений/снять защиту рабочего листа
Select	Выбрать рабочий лист

В табл. В3 приведены основные методы для работы с диапазоном ячеек.

Таблица В3. Методы для работы с диапазоном ячеек

Название метода	Назначение метода
Activate	Активировать одну ячейку в текущем выделении
AddComment	Добавить комментарий к диапазону (ячейке)
ClearComments	Удалить все комментарии к ячейкам заданного диапазона
AutoFill	Выполнить автозаливку ячеек заданного диапазона
AutoFilter	Отфильтровать список автофильтром
BorderAround	Задать обрамляющие границы диапазона
Clear	Удалить формулы и форматирование в ячейках
ClearContents	Очистить формулы, оставить форматирование
ClearFormats	Очистить форматирование
Copy	Копировать в заданный диапазон либо в буфер обмена
CopyPicture	Копировать объект в буфер обмена в виде рисунка
Delete	Удалить (можно со сдвигом)
Find	Найти указанную информацию
FindNext	Найти следующее
FindPrevious	Найти предыдущее
FunctionWizard	Запустить мастер функций для верхней левой ячейки
Insert	Вставить
PasteSpecial	Специальная вставка
Replace	Заменить
Select	Выбрать ячейку или диапазон ячеек
Show	Отобразить диапазон из одной ячейки

События

Событие (event) — это распознаваемое объектом действие, для которого можно запрограммировать отклик. События бывают разные: это и открытие, и сохранение, и закрытие проекта. Это и щелчок левой или правой кнопкой мыши, удержание нажатия кнопки и перетаскивание курсора и т. д.

События могут быть инициированы как пользователями компьютера или его программой, так и операционной системой.

Событие рассматривается с двух сторон: как действие и как отклик на это действие. В этом и состоит суть программирования на VBA. Например, при щелчке по кнопке выполняется код процедуры, созданной программистом. Если программа не написана или написана с ошибками, то не будет и отклика на событие. Специальный вид процедур, генерирующих отклик на событие, называется *процедурами обработки событий*.

Можно сказать, что программирование на VBA состоит в создании кода программ, которые генерируют прямо или косвенно отклики на события.

В заключение можно отметить, что VBA — это интерпретатор, он не создает исполнимых exe-файлов. Сам код в виде текста (Юникод) хранится в том же документе, созданном приложением, в которое встроен VBA. Макросы выполняются достаточно медленно, но в большинстве случаев для автоматизации документооборота большая скорость не требуется.



ГЛАВА 1

Основные понятия VBA

Базовые конструкции языка Visual Basic

Начиная изучение языка программирования *Microsoft Visual Basic for Applications* (VBA), рассмотрим сначала такое понятие, как язык. Язык — это знаковая система, предназначенная для хранения и передачи информации. Причем неважно, естественный это язык или язык программирования (а может, математический или язык жестов).

Безусловно, языки различаются между собой, однако большинству языков присущи общие закономерности и системная организация. Для каждого языка определены следующие понятия:

- ◆ *алфавит*, определяющий допустимые символы языка;
- ◆ *лексика* — словарь языка и способы образования слов из символов;
- ◆ *синтаксис* — способы и правила записи и соединения слов в предложения. В VBA это правила записи лексем языка: констант, имен переменных, массивов, ключевых слов, строк и др., а также способы соединения слов пользователя, знаков операций и ключевых слов VBA в различные программные конструкции;
- ◆ *семантика* — значения отдельных слов, словосочетаний и предложений. В VBA этот термин определяет значение и логический смысл выражений и записываемых пользователем программных конструкций;
- ◆ *прагматика* — назначение и область применения языка.

Как и все языки программирования высокого уровня, Visual Basic имеет свой набор допустимых для использования символов — алфавит. Алфавит VBA содержит следующие группы символов:

- ◆ *буквы*: заглавные и строчные буквы латинского алфавита и кириллицы — A, B, ..., Z, a, b, ..., z, А, Б, ..., Я, а, б, ..., я;
- ◆ *арабские цифры*: 0, 1, ..., 9;

◆ *специальные символы:*

+ - * / \ ' " . , = > <
[] () : { } & @ % \$! ;

◆ *разделители:* пробел, символ продолжения строки (`
`), символ табуляции, символ новой строки.

Из символов алфавита конструируются все структуры языка, в том числе составные символы (`<=`, `<>`, `>=`), ключевые слова (`Dim`, `As`, `Binary` и т. п.), а также имена констант, переменных, процедур, функций и т. п.

Заглавные и строчные буквы в записи имен VBA не различаются. Транслятор запоминает формат первой записи идентификатора в программе, и при повторном использовании идентификатора в тексте программы он автоматически преобразуется к первоначальному формату.

Типы данных в VBA

Тип данных программного элемента определяет данные, которые могут содержаться в нем, и способы их хранения. Типы данных применяются ко всем значениям, которые могут храниться в памяти компьютера или участвовать в вычислении выражения.

Каждая переменная, литерал, константа, перечисление, свойство, параметр процедуры, аргумент процедуры и возвращаемое значение процедуры имеют тип данных.

Переменные подразделяются на *простые* и *индексированные* (переменные с индексом). Индексированными переменными являются элементы массивов. Все подробности о массивах будут рассмотрены позднее.

Все слова, задействованные в конструкциях языка, являются зарезервированными словами и не могут применяться для других целей.

В языке программирования Visual Basic версии для Microsoft Excel 2019 определены различные типы данных (табл. 1.1), используемые как напрямую, так и в универсальных типах данных (например, `Decimal` в `Variant`): `Boolean`, `Byte`, `Currency`, `Date`, `Double`, `Decimal`, `Integer`, `Long\LongLong`, `LongPtr`, `Object`, `Single`, `String`, `Variant`, а также типы данных, определяемые пользователем так же, как некоторые типы объектов.

Таблица 1.1. Типы данных, используемые в программах на Visual Basic для Excel 2019

Тип	Занимаемый объем памяти	Диапазон значений
<code>Boolean</code>	2 байта	Логическая величина, имеющая значения <code>True</code> (Истина) или <code>False</code> (Ложь)
<code>Byte</code>	1 байт	Целое число без знака в диапазоне от 0 до 255
<code>Currency</code> (денежный)	8 байтов	Число с фиксированной десятичной точкой от -922 337 203 685 477,5808 до 922 337 203 685 477,5807

Таблица 1.1 (окончание)

Тип	Занимаемый объем памяти	Диапазон значений
Date	8 байтов	Дата от 1 января 100 года до 31 декабря 9999 года. Время от 0:00:00 до 23:59:59
Double	8 байтов	Число с плавающей запятой двойной точности от $\pm 1,79769313486231570E+308$ до $\pm 4,94065645841246544E-324$
Decimal	14 байтов	От 0 до $\pm 79\,228\,162\,514\,264\,337\,593\,543\,950\,335$ ($\pm 7,9E+28$) без десятичной запятой; от 0 до $\pm 7,9228162514264337593543950335$ с 28 разрядами справа от десятичной запятой; наименьшее ненулевое число — это $\pm 0,00000000000000000000000000000001$ ($\pm 1E-28$). Тип данных используется только в элементе Variant
Integer	2 байта	Целое число от $-32\,768$ до $32\,767$
Long (целое число Long)	4 байта	Целое число от $-2\,147\,483\,648$ до $2\,147\,483\,647$
LongLong (целое число LongLong)	8 байтов	Тип данных предназначен для 64-разрядных систем. Целое число от $-9\,223\,372\,036\,854\,775\,808$ до $9\,223\,372\,036\,854\,775\,807$
LongPtr	4 байта/8 байтов	Ненстоящий тип данных, преобразуется в целое число типа Long в 32-разрядных системах либо в целое число типа LongLong в 64-разрядных
Object	4 байта	Любая ссылка на объект VBA
Single	4 байта	Число с плавающей запятой одинарной точности от $\pm 3,402823E38$ до $\pm 1,401298E-45$
String	а) 10 байтов + длина строки; б) длина строки	а) переменной длины — до 2^{31} символов; б) фиксированной длины — до 64K (2^{16}) символов
Variant	а) строковое зна- чение: 22 байта + длина строки; б) числовое значе- ние: 16 байтов	Тип данных, определяемый по умолчанию для переменных, не объявленных явно: а) любое числовое значение до диапазона типа Double; б) диапазон как для типа String переменной длины
Type	В зависимости от числа элементов	Тип данных, определяемый пользователем. Диапазон каждого элемента совпадает с диапазоном его типа данных

В экспоненциальном представлении символ E обозначает основание степени 10. Поэтому $3,56E+2$ означает $3,56 \cdot 10^2$ или 356, а $3,56E-2$ означает $3,56/10^2$ или 0,0356.

Константы и переменные, переменные объектов

В VBA для хранения значений предусмотрены просто переменные, переменные объектов и константы.

- ◆ *Константа* — именованная область памяти, используемая для хранения фиксированного значения, не изменяемого при выполнении программы.
- ◆ *Переменная* — именованная область памяти, используемая для хранения значения, которое можно изменить при выполнении программы.
- ◆ *Переменная объектов* — именованная область памяти, используемая для хранения любого объекта, используемого в Microsoft Excel (ячейка, диапазон ячеек, рабочий лист и т. д.), над которым выполняются действия при работе программы. Переменные объектов используются для ускорения работы программы и более наглядного кода.

Область действия переменных и констант

Область действия переменных и констант определяется с помощью ключевых слов `Private`, `Public` и `Static`.

- ◆ `Private` — определяет область действия в пределах конкретного модуля программы. После завершения выполнения модуля память, отведенная под эти переменные, освобождается. Переменные и константы, объявленные как `Private`, сохраняют свои значения только во время выполнения блока кода, в котором они определены.
- ◆ `Public` — область действия в пределах приложения. Переменные и константы, объявленные как `Public`, сохраняют свои значения до конца выполнения программы.
- ◆ `Static` — область действия в пределах конкретного модуля и внешних процедур, задействованных в этом модуле. После завершения выполнения модуля значения этих переменных сохраняются между вызовами процедур и могут быть использованы при повторном выполнении модуля.

Переменные и константы, объявленные без ключевых слов `Public`, `Private`, `Static`, сохраняют свое значение согласно месту их объявления (описания).

Переменные и константы в зависимости от области действия подразделяются на глобальные и локальные.

- ◆ Если переменная или константа объявлена внутри процедуры, то она является *локальной*, т. е. она определена и доступна только в пределах данной процедуры.
- ◆ Если переменная или константа объявлена вне процедуры, то она будет *глобальной*. Такая переменная или константа может быть доступна в нескольких процедурах.