

Андрей Гарнаев



VBWA

- Интеграция офисных приложений
- Применение XML и веб-служб
- Программирование MS Agent и смарт-тегов
- Обработка и анализ данных
- Управление диаграммами и создание нестандартных диаграмм
- Около 500 примеров

**Наиболее
полное
руководство**

В ПОДЛИННИКЕ®

Андрей Гарнаев

VBA

Санкт-Петербург

«БХВ-Петербург»

2005

УДК 681.3.06+800.92VBA
ББК 32.973.26-018.1
Г20

Гарнаев А. Ю.

Г20 VBA.— СПб.: БХВ-Петербург, 2005. — 848 с.: ил.
ISBN 5-94157-633-1

Рассмотрены многочисленные средства и возможности языка VBA для повышения гибкости и расширения функциональности офисных приложений Excel, Word, Access, PowerPoint и Outlook. Описан синтаксис языка и даны практические рекомендации по объектно-ориентированному программированию. Показано, как автоматизировать анализ и обработку данных, решать оптимизационные задачи и строить стандартные и нестандартные диаграммы в Excel; как работать со стилями, шаблонами и формами в Word; как создавать интерактивные средства по работе с базами данных в Access. Большое внимание уделено конструированию пользовательского интерфейса как на этапе дизайна, так и во время выполнения приложения. Рассмотрены вопросы взаимодействия офисных приложений между собой, а также их интеграция в .NET-проекты. Показано, как подсоединить к приложениям Web-ресурсы с помощью Web-служб и Web-запросов. Книга содержит около 500 тщательно разработанных примеров, которые могут быть использованы для создания собственных приложений.

Для пользователей и программистов

УДК 681.3.06+800.92VBA
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Наталья Першакова</i>
Дизайн обложки	<i>Игоря Цырульниковой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 30.09.05.
Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 68,37.
Тираж 3000 экз. Заказ №
"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-633-1

© Гарнаев А. Ю., 2005
© Оформление, издательство "БХВ-Петербург", 2005

Оглавление

Предисловие	1
О чем эта книга?	1
Для чего написана эта книга?	3
Структура книги	4
Для кого предназначена эта книга?	8
Об авторе.....	9
Глава 1. Введение в VBA	11
Макросы.....	11
Запись макроса.....	12
Выполнение макроса	17
Редактирование макроса	17
Назначение макроса объекту	21
Распространение макросов	22
Защита от макросов	22
Элемент управления <i>Кнопка</i> и создание процедуры обработки события, при которой выполняется макрос.....	23
Назначение макроса кнопке пользовательского меню.....	26
Управление диаграммой	28
Создание скриптов для Windows.....	30
Создание скриптов для Web на стороне клиента.....	31
Создание скриптов для Web на стороне сервера.....	33
Передача данных клиенту из базы данных на сервере	36
Добавление анимационных эффектов в форму Access	37
Глава 2. Интегрированная среда разработки	39
Где набирается код VBA?	39
Структура редактора VBA.....	40
Окно <i>Project - VBAProject</i>	40
Копирование модулей и форм из одного проекта в другой.....	41
Окно редактирования кода	41
Допустимый объем модуля.....	43
Интеллектуальные возможности редактора кода	43
Окно <i>UserForm</i> (Редактирование форм)	45
Окно <i>Properties</i> (Свойства).....	48
Окно <i>Object Browser</i> (Просмотр объектов)	49
Программирование среды разработки	51
Экспортирование модуля	52
Удаление модуля.....	52

Программа, которая выполняется только один раз	53
Импорт данных в модуль из текстового файла.....	53
Очистка содержания модуля	54
Проверка существования модуля или процедуры	54
Программное создание кода	55
Удаление кода процедуры из модуля	56
Список всех процедур модуля	57
Создание пользовательской формы	57
Построение формы с элементом управления, реагирующим на события.....	58
Глава 3. Элементы ООП в VBA	61
Классы и экземпляры	61
Объявление класса.....	62
Создание экземпляра класса.....	62
Инициализация значений полей	63
Методы и свойства.....	64
События.....	67
Интерфейсы	70
Объект <i>Collection</i>	71
Построение стека	74
Глава 4. Работа со строками, временем и датами	77
Функции обработки строк	77
Нахождение ASCII-кода по литералу и литерала по ASCII-коду.....	77
Преобразование строки к нижнему или верхнему регистру	78
Преобразование первых букв слов в прописные	78
Возвращение подстроки по указанному местоположению	78
Как узнать, содержит ли строка число?	79
Определение длины строки.....	79
Копия строки без начальных и конечных пробелов.....	79
Строка, состоящая из указанного числа пробелов.....	80
Строка, состоящая из указанного числа повторяющихся символов.....	80
Преобразование строки в массив	81
Преобразование массива в строку	82
Сравнение строк.....	82
Создание зеркальной строки по отношению к данной	83
Нахождение вхождения в строку подстроки.....	83
Замена в строке подстроки	83
Проверка, является ли строка электронным адресом.....	84
Функции даты и времени	85
Определение текущей системной даты	86
Определение текущего системного времени.....	86
Определение текущей даты и системного времени.....	86
Извлечение из времени часового, минутного и секундного компонентов	86

Извлечение из даты годового, месячного и дневного компонентов.....	87
Определение дня недели	87
Определение числа секунд, прошедших с полуночи	87
Определение количества лет, кварталов, месяцев, недель и дней, прошедших между двумя датами	88
Определение компонента даты	90
Добавление к дате указанного временного интервала	90
Преобразование часов, минут и секунд в формат времени.....	91
Преобразование года, месяца и дня в формат даты.....	91
Преобразование строки в формат времени	91
Проверка, является ли год високосным	92
Определение локальных установок.....	92

Глава 5. VBA и MS Excel..... 95

Объектная модель.....	95
Полная и неявная ссылка на объект.....	95
Объект <i>Application</i>	97
Свойства объекта <i>Application</i>	97
Ссылка на активную рабочую книгу, лист, ячейку, диаграмму и принтер.....	98
Загрузка инсталлированной надстройки.....	99
Управление уровнем безопасности.....	100
Рабочая книга, в которой выполняется данный макрос.....	101
Установка заголовка окна MS Excel.....	101
Работа со строкой состояния	101
Отключение генерации событий.....	102
Установка типа курсора.....	102
Семейство встроенных диалоговых окон.....	103
Объект <i>FileDialog</i>	104
Отображение встроенных предупреждений о работе программы.....	106
Поиск файлов	106
Печать активного листа	108
Предварительный просмотр книги.....	108
Доступ из кода к функциям рабочего листа.....	108
Номер версии MS Excel.....	109
Определение локальной версии MS Excel и установок Windows.....	109
Методы объекта <i>Application</i>	110
Установка выполнения специфицированной процедуры при нажатии заданной комбинации клавиш.....	110
Переопределение горячих клавиш приложения	112
Проверка правописания отдельного слова.....	113
Преобразование имени MS Excel в объект или значение.....	113
Симулирование вычисления арифметических выражений.....	114
Назначение выполнения процедуры на определенное время	115
Закрытие приложения.....	117
Сохранение изменений во всех рабочих книгах	118

Приостановка работы приложения до указанного времени.....	118
Получение данных из закрытой рабочей книги	119
События объекта <i>Application</i>	119
Семейство <i>Workbooks</i>	122
Создание новой рабочей книги	122
Открытие рабочей книги	122
Объект <i>Workbook</i>	123
Свойства объекта <i>Workbook</i>	124
Закрытие книги без сохранения изменений.....	128
Определение объекта <i>Workbook</i> по имени рабочей книги.....	128
Как узнать, открыта ли специфицированная рабочая книга	128
Объект <i>Name</i> и простой способ удаления из рабочей книги ненужных имен.....	129
Определение, была ли сохранена открытая рабочая книга.....	130
Методы объекта <i>Workbook</i>	130
Установка и снятие защиты книги	132
Сохранение рабочей книги с именем, соответствующим текущей дате.....	133
События объекта <i>Workbook</i>	134
Запрет закрытия рабочей книги.....	135
Автоматическое изменение размеров столбцов при проведении вычислений	135
Как отследить за тем, кто открывал рабочую книгу?.....	136
Семейство <i>Worksheets</i>	136
Вставка нового листа с именем, отличным от существующих	138
Объект <i>Worksheet</i>	139
Свойства объекта <i>Worksheet</i>	139
Проверка установки защиты на содержание рабочего листа	143
Проверка существования рабочего листа	143
Методы объекта <i>Worksheet</i>	144
Удаление рабочего листа без предупреждения пользователя	145
Установка и снятие защиты с рабочего листа.....	145
Объект <i>Protection</i>	146
Объект <i>PageSetup</i> и как вывести в колонтитул имя книги, рабочего листа и текущую дату.....	149
События объекта <i>Worksheet</i>	149
Блокировка действий, связанных с событием по умолчанию.....	150
Добавление к графическому объекту обработчика события <i>Click</i>	151
Автоматическое переоформление таблицы при изменении в ней значений	152
Условное форматирование без применения кода.....	154
Отображение результата, только при условии ввода всех данных.....	155
Проверка вводимых значений.....	155
Автоматический ввод данных в верхнем регистре данного диапазона	156
Вывод суммы значений из выделенного диапазона в строку состояния.....	157

Объекты <i>Range</i> и <i>Selection</i>	157
Адресация ячеек	158
Задание групп строк и столбцов	159
Связь объекта <i>Range</i> и свойства <i>Cells</i> объекта <i>Worksheet</i>	159
Свойства объекта <i>Range</i>	160
Ввод или считывание значения из диапазона	161
Поиск по шаблону подобных значений в диапазоне	161
Ввод или считывание формулы в ячейку в формате <i>A1</i>	162
Ввод или считывание формулы в ячейку в формате <i>R1C1</i>	162
Ввод или считывание формулы локальной версии в ячейку в формате <i>A1</i>	162
Ввод или считывание формулы локальной версии в ячейку в формате <i>R1C1</i>	162
Ввод формулы массива в диапазон	162
Ввод формулы массива локальной версии в диапазон	163
Ввод формулы массива в диапазон с относительными ссылками на ячейки	163
Определение адреса ячейки	163
Управление стилем границы диапазона и объекта <i>Border</i>	164
Функции <i>RGB</i> и <i>QBColor</i>	165
Доступ к отдельным ячейкам диапазона	167
Выбор элементов на рабочем листе или в книге	168
Как проверить, является ли диапазон выбранным объектом?	170
Определение типа данных, хранимых в ячейке	170
Объект <i>Characters</i>	171
Объект <i>Comment</i>	173
Определение текущего диапазона	176
Нахождение крайней ячейки диапазона в указанном направлении	176
Нахождение строки и столбца, содержащих данную ячейку	176
Объект <i>Hyperlink</i>	177
Объект <i>Font</i>	179
Объект <i>Interior</i>	180
Разрешение редактирования содержимого ячеек на защищенном рабочем листе	181
Установка числового формата	182
Нахождение диапазона, сдвинутого относительно данного на указанное число строк и столбцов	182
Задание угла, под которым выводится текст в диапазоне	183
Переопределение размеров диапазона	183
Методы объекта <i>Range</i>	184
Активизация и выбор диапазона	184
Вставка и удаление комментариев в диапазон	185
Заполнение диапазона прогрессией	185
Автозаполнение ячеек диапазона элементами последовательности	187
Табуляция функций	188
Заполнение диапазона по одному значению	190

Очистка ячейки.....	190
Копирование, вырезание и удаление данных из диапазона.....	190
Специальная вставка.....	191
Вставка диапазона с транспонированием.....	192
Копирование диапазона в буфер обмена как растровое изображение.....	192
Бегущая картинка.....	193
Поиск значения в диапазоне.....	194
Повторный поиск и поиск всех значений.....	196
Поиск по формату.....	196
Замена значений.....	198
Подбор параметра и решение уравнения с одной неизвестной.....	199
Ввод в диапазон неповторяющихся значений.....	201
Озвучивание текста.....	202
Построение графа.....	203
Пользовательские функции.....	205
Математические функции.....	206
Надстройки.....	207
Где хранятся макросы?.....	207

Глава 6. Формы и элементы управления.....209

Первая форма.....	210
Форма как объект.....	210
Отображение формы при выборе ячейки рабочего листа.....	212
Запуск проекта на исполнение.....	213
Ключевое слово <i>Me</i>	213
Создание копии формы.....	214
Закрытие сразу всех форм.....	214
Предотвращение закрытия окна с помощью кнопки <i>Close</i>	215
Вывод формы без отображения приложения.....	215
Надпись.....	216
Добавление на форму гиперссылки.....	217
Всплывающая форма.....	218
Кнопка.....	220
Кнопка с горячей клавишей.....	222
Настройка панели инструментов <i>Toolbox</i>	223
Массивы элементов управления.....	224
Поле.....	225
Интерфейс быстрого ввода.....	227
Автоматический перевод вводимых числовых данных в денежный формат.....	227
Как полю получить фокус и в нем выделить весь текст или его часть?.....	229
Многострочное поле.....	229
Поле ввода пароля.....	231
Всплывающая подсказка.....	232
Привязка контекстного меню к элементу управления.....	232

Список.....	235
Выбор нескольких элементов из списка	236
Динамический вывод данных в список по критерию.....	238
Передача значения форме.....	240
Поле со списком	243
Очистка в одном цикле всех полей и полей со списками.....	245
Привязка поля со списком к данным из рабочего листа.....	245
Рисунок	249
Вставка в рисунок изображения выделенного диапазона	250
Загрузка в список имен всех рисунков указанного каталога	252
Флажок.....	252
Использование списка вместо группы флажков	254
Рамка	255
Переключатель.....	255
Запуск и остановка часов при помощи переключателя.....	257
Выключатель.....	258
Полоса прокрутки	260
Счетчик	261
Обеспечение функционирования кнопки только в случае заполненных полей	264
Создание элементов управления в коде	265
Добавление массива кнопок в коде	269
Добавление элементов управления на рабочий лист в коде	270
Добавление элементов управления из панели инструментов <i>Формы</i>	273
Добавление кнопки в документ.....	274
Объект <i>Data Object</i>	274
Применение стандартных форм при просмотре данных, введенных на рабочем листе	277
Дополнительные элементы управления.....	278
Пример дополнительного элемента управления — <i>Календарь</i>	279
Назначение одного обработчика событий нескольким элементам управления	281

Глава 7. Создание меню, контекстного меню и панели инструментов..... 283

Создание строки меню.....	283
Типичные ошибки, возникающие при создании пользовательских панелей инструментов	287
Отображение имен всех меню	287
Пункты меню со встроенными функциями и картинками.....	288
Отображение всех встроенных значков в меню	291
Меню с флажками.....	292
Добавление нового элемента в существующее меню.....	294
Создание панели инструментов.....	295
Расположение нескольких панелей инструментов в один ряд	302
Добавление раскрывающегося списка в панель инструментов	302
Поле ввода в меню.....	305

Добавление шпиона в контекстное меню	306
Конструирование контекстного меню	307
Связывание меню с элементом управления, расположенным на форме	309
Глава 8. Диаграммы	311
Построение диаграммы	311
Получение ссылки на активную диаграмму	313
Изменение типа рядов данных	313
Построение внедренной диаграммы	315
Установка размеров и местоположения внедренной диаграммы	317
Удаление диаграммы	317
Добавление на диаграмму графических элементов	320
Сохранение диаграммы как графического файла	322
Задание цветов серий	323
Печать и предварительный просмотр печати диаграммы	325
Построение диаграммы на основе данных из нескольких листов	325
Защита диаграммы	327
Изменение диапазона, на основе которого конструируется диаграмма	328
Еще раз об изменяемом диапазоне	330
Диаграмма в полярных координатах	332
Добавление рисунка	334
Расположение нескольких диаграмм на одном листе	335
График выполнения работ	336
Диаграмма Гранта	339
Диаграммы линейного и нелинейного тренда	340
Диаграмма сравнения	343
Построение графика без табулирования	344
График, зависящий от параметра	346
События и диаграммы	347
Привязка событий к вложенным в рабочий лист диаграммам	348
Изменение типа диаграммы при помощи контекстного меню	349
Отображение диаграммы в форме	352
Глава 9. Помощник MS Office	355
Объект <i>Assistant</i>	355
Информирование об изменениях в приложении с помощью помощника	357
Объект <i>Balloon</i>	358
Помощник с надписями	360
Помощник с флажками	362
Немодальное окно помощника	363
Microsoft Agent	365
Агент, читающий текст из поля	369
Контекстное меню	370
Синхронизация действий и речи	372
Мультиагентные приложения	372

Глава 10. Работа с файлами	375
Объект <i>FileSystemObject</i>	375
Создание объекта <i>FileSystemObject</i> с явной ссылкой на Microsoft Scripting Runtime.....	375
Создание объекта <i>FileSystemObject</i> с неявной ссылкой на Microsoft Scripting Runtime.....	376
Проверка существования диска	376
Получение информации о диске	377
Проверка существования каталога или файла	378
Создание и удаление каталога.....	379
Получение информации о каталоге.....	380
Копирование и перемещение файла	381
Удаление файла.....	382
Информация о файле.....	382
Список всех файлов данного каталога.....	383
Открытие файла и создание <i>TextStream</i> -объекта.....	383
Метод <i>OpenAsTextStream</i>	384
Метод <i>CreateTextFile</i>	385
Метод <i>OpenTextStream</i>	386
Запись, присоединение и считывание данных из файла	387
Функции по работе с файлами	389
Просмотр файлов в каталоге.....	390
Поиск файла и объект <i>FileSearch</i>	391
Определение существования рабочей книги.....	393
Глава 11. Обработка ошибок и отладка программ	395
Разработка процедур, предотвращающих появление ошибок	395
Контроль вводимых значений с помощью обработки события <i>KeyPress</i>	398
Перехват и обработка ошибок	399
Оператор <i>On Error</i>	400
Процедура обработки ошибки	400
Оператор <i>Resume</i>	400
Объект <i>Err</i>	401
Создание пользовательских ошибок для тонкой настройки обработчика ошибок.....	405
Отображение ошибок при работе с пользовательскими функциями.....	406
Отладка программ	406
Ошибки компиляции	407
Ошибки выполнения	408
Логические ошибки	410
Инструкция <i>Option Explicit</i>	410
Пошаговое выполнение программ	410
Точка прерывания.....	411

Вывод значений свойств и переменных	412
Окно <i>Watches</i>	413
Окно <i>Locals</i>	413
Окно <i>Immediate</i>	413
Программный способ вывода значений в окно <i>Immediate</i>	414
Глава 12. VBA и Win32 API.....	415
Первый пример	415
Отправка электронной почты и загрузка Web-страницы	416
Определение и изменение разрешения экрана.....	418
Открытие окна Проводника	418
Анимация	419
Очистка корзины.....	420
Определение пуст ли буфер обмена	420
Отображение окна свойств специфицированного файла.....	421
Загрузка Web-страницы	422
Действия с окном.....	423
Форма произвольной конфигурации	424
Форма без заголовка	428
Печать формы.....	429
Замена обоев.....	431
Глава 13. Обработка данных	433
Консолидация данных.....	433
Консолидация при помощи трехмерных формул на рабочем листе.....	434
Консолидация при помощи трехмерных формул в коде.....	435
Консолидация данных по положению и категориям.....	436
Методы и свойства, используемые при программировании консолидирующей таблицы	438
Пример приложения, консолидирующего данные	440
Сортировка данных.....	444
Сортировка по выделенному полю	444
Сортировка выделенной области.....	444
Сортировка по двум или трем столбцам	444
Сортировка по списку или по дням недели.....	445
Сортировка с учетом регистра.....	446
Восстановление первоначального порядка вывода данных.....	446
Сортировка на защищенном листе	446
Метод <i>Sort</i>	447
Пример приложения, сортирующего данные	448
Фильтрация.....	451
Использование автофильтра	452
Процесс автофильтрации.....	452
Метод <i>AutoFilter</i>	454
Отображение и удаление кнопок со стрелками автофильтра.....	455
Отображение всех отфильтрованных данных.....	455

Как узнать, что на листе имеются кнопки со стрелками автофильтра?	455
Как узнать, что на листе имеются отфильтрованные данные?	455
Объекты <i>AutoFilter</i> и <i>Filter</i> и семейство <i>Filters</i>	455
Доступ к раскрывающимся спискам автофильтра на защищенном листе	457
Пример приложения, фильтрующего данные	457
Промежуточные итоги	461
Создание промежуточных итогов	461
Метод <i>Subtotal</i>	464
Удаление промежуточных итогов	465
Структура и объект <i>Outline</i>	465
Отображение указанного числа уровней структуры	466
Удаление структуры	466
Отображение значков структуры	466
Автоматическое создание структуры	467
Пример приложения, подводящего промежуточные итоги и управляющего структурой	467
Сценарии	469
Объект <i>Scenario</i>	469
Пример приложения по работе со сценариями	471
Сценарии без объекта <i>Scenario</i>	473
Сводная таблица	474
Подведение итогов по клиентам	475
Обновление данных	482
Подведение итогов по клиентам и странам	483
Сортировка данных в сводной таблице	484
Отображение заданного числа лучших или худших результатов	485
Группировка данных по дням, месяцам, кварталам и годам	486
Удаление группировки	488
Удаление сводной таблицы	489
Отображение детальных данных, содержащихся в ячейке сводной таблицы	489
Объекты, связанные со сводной таблицей	489
Метод <i>PivotTableWizard</i>	490
Объект <i>PivotTable</i>	491
Объект <i>PivotCache</i>	492
Объект <i>PivotField</i>	493
Пример приложения, помогающего построить и обновлять сводную таблицу	494
Списки	501
Поиск решения	504
Решение транспортной задачи при помощи средства Поиск решения	504
Функции, программирующие поиск решения	511
Приложение "Транспортная задача"	514

Глава 14. VBA и MS Word	519
Создание стилей.....	519
Программное задание стиля.....	521
Создание шаблонов.....	522
Создание документа по шаблону.....	523
Присоединение шаблона к существующему документу	523
Вставка комментариев	523
Вставка специального символа.....	525
Использование закладок для заполнения письма.....	526
Использование форм	528
Создание WordArt-объекта на основе выделенного текста	532
Добавление кнопки в новый документ.....	532
Удаление повторяющегося текста.....	533
Преобразование форматов дат.....	535
Замена цифр комментариями	535
Отображение стандартных окон при выполнении замен текста	536
Преобразование гиперссылок.....	536
Отображение статистики в заголовке приложения	537
Ввод данных в таблицу.....	538
Создание таблицы из структурированного текста.....	539
Ускорение обработки текстовых данных.....	540
Импорт данных из базы данных.....	540
Хранение в документе скрытой информации.....	541
Глава 15. VBA и Automation	543
Средство Automation.....	543
Программные идентификаторы приложений — серверов Automation	544
Функции доступа к объектам Automation.....	544
Позднее и ранее связывание.....	545
MS Word	546
Открытие Word-документа.....	546
Создание Word-документа.....	546
Создание Word-отчета на основе данных, хранимых на рабочем листе.....	547
Копирование диапазона данных из рабочего листа в Word-документ.....	551
MS Excel	551
Создание презентации на основе данных, хранимых на рабочем листе MS Excel.....	551
MS PowerPoint	555
Копирование диапазона данных из рабочего листа в таблицу PowerPoint	555
Копирование выбранной диаграммы в текущий слайд активной PowerPoint-презентации	556
MS Outlook.....	557
Получение списка всех контактов в Outlook	557
Копирование всех выделенных сообщений из Outlook в Word-документ.....	558

Глава 16. ActiveX Data Objects	561
Создание ссылки на библиотеку ADO	562
Объект <i>Connection</i> и установка подключения к базе данных	563
Установка DSN.....	564
Проверка установления соединения.....	565
Открытие защищенной базы данных.....	565
Объект <i>Recordset</i> и его создание	566
Пример использования объекта <i>Recordset</i> для просмотра базы данных.....	566
Работа с курсорами	568
Создание объекта <i>Recordset</i> методом <i>Execute</i>	569
Методы, свойства и события объекта <i>Recordset</i>	570
Как проверить, не пуст ли объект <i>Recordset</i> ?	573
Последовательный просмотр записей базы данных.....	574
Доступ к данным, расположенным на рабочих листах закрытой книги	577
Объект <i>Field</i>	578
Создание браузера базы данных.....	579
Редактирование, создание, обновление и удаление записей	582
Создание закладок	587
Поиск записи методом <i>Find</i>	588
Использование объекта <i>Command</i>	591
Транзакция.....	592
Глава 17. VBA и MS Access.....	595
Макросы в Access.....	595
Применение объекта <i>DoCmd</i>	596
Выполнение команды встроенного меню и блокировка отображения предупреждений	597
Блокировка пользовательского доступа к объектам Access при закрытии формы	597
Объектная модель MS Access.....	597
Открытие и закрытие MS Access из других приложений	598
Ваша первая форма в MS Access.....	599
Создание всплывающей формы	601
Анимация с помощью Microsoft Animation Control	603
Выбор проигрываемого анимационного файла.....	604
Открытие одной формы из другой.....	605
Присоединение меню к форме.....	606
Создание формы без меню и панели инструментов	606
Обращение к объекту по имени	607
Создание запроса	608
Вызов запроса из формы.....	610
Источники данных.....	612
Извлечение данных из списка в несвязанный элемент управления.....	613
Программная навигация по записям	615
Назначение одного обработчика событий группе элементов управления.....	618

Определение активного объекта.....	620
Объект <i>CurrentData</i>	620
Объект <i>CurrentProject</i>	620
Кнопка с двумя рисунками в зависимости от ее состояния	621
Список таблиц и полей	622
Создание расширяющегося диалогового окна.....	623
Создание копии формы.....	625
Использование ADO в Access	627
Экспорт данных в форму MS Word	628

Глава 18. VBA и Web 631

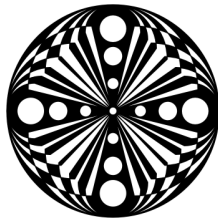
Отправка сообщений по электронной почте	631
Задание параметров Web-страницы	632
Предварительный просмотр документа как Web-страницы.....	633
Сохранение документа как Web-страницы	633
Объект <i>PublishObject</i>	633
Гиперссылки	635
Отправка сообщения с помощью гиперссылки.....	635
Открытие всех гиперссылок, отмеченных флажками	636
Открытие документа	637
Объект <i>InternetExplorer</i>	637
Поиск всех гиперссылок Web-страницы с помощью объекта <i>InternetExplorer</i>	637
Создание объекта <i>InternetExplorer</i> функцией <i>CreateObject</i>	638
Определение загрузки Web-страницы.....	639
Получение точного времени из сети.....	640
Получение доступа к тегу документа	641
Чтение содержимого буфера обмена.....	641
Определение разрешения экрана	642
Объект <i>HTMLDocument</i>	642
Элемент управления <i>WebBrowser</i>	643
Печать содержимого Web-страницы.....	646
Отображение анимированных GIF-файлов	646
Просмотр презентации на форме	646
Вставка рисунка в рабочий лист из Web	647
Загрузка файла из сети	648
Создание Web-запроса.....	649
Получение данных из Интернета.....	652
Получение данных из Интернета по запросу из указанной таблицы	653
Web-службы	654
Создание Web-службы с помощью Visual Studio .NET.....	654
Применение Web-службы в MS Office.....	657
Опубликование данных в Интернете из Access с помощью страниц доступа к данным	660
Создание страницы доступа, содержащей все поля и записи таблицы.....	661
Мастер страниц	662

Глава 19. VBA и Windows-сценарии	663
Первый Windows-сценарий	663
Передача значений сценарию через параметры файла	664
Вызов сценария из офисного приложения	665
Доступ к файловой системе	665
Доступ к специальным папкам	666
Замена указанных префиксов имен файлов данного каталога	667
Завершение работы с Windows	668
Создание документа Word и его печать	668
Доступ к открытому активному Word-документу	669
Как определить, запущено ли приложение?	669
Считывание значения из ячейки рабочей книги	670
Перевод HTML-документа в RTF-формат	670
Использование MS Excel в сценариях	671
Доступ к программам и передача в них данных с клавиатуры	673
Запуск Internet Explorer из скрипта	674
Ввод и вывод данных в консольное окно	674
Запуск DOS-команды	675
Временные диалоговые окна	675
Создание ярлыка	676
Запуск MS Agent из скрипта	677
Использование классов в сценариях	678
Глава 20. Microsoft Visual Studio Tools для MS Office	681
Ваш первый проект в Microsoft Visual Studio Tools	682
Перехват событий рабочего листа	684
Загрузка данных из базы данных на рабочий лист MS Excel	685
Создание меню	686
Заполнение таблицы в Word	689
Microsoft Visual Studio Tools и Web-службы	691
Глава 21. VBA и XML	695
Синтаксис XML-документа	695
Синтаксис XSD-документа	698
Открытие XML-документа в Excel без использования схемы	701
Открытие XML-документа в Excel с использованием схемы	702
Импорт XML-карты в коде	703
Создание связанных с картой столбцов и импорт в них данных	703
Отслеживание и сохранение в формате XML ежедневных изменений в документе	705
XSL	706
Библиотека MSXML	708
Перебор узлов	710
Атрибуты	713
Создание узла	715

Глава 22. VBA и .NET	717
Создание DLL-файлов в .NET без использования шаблона	717
Создание DLL-файлов в .NET с использованием шаблона.....	719
Встроенные процедуры.....	720
Нахождение значений арифметических выражений	722
Проверка орфографии	724
Создание отчета в MS Word.....	726
Построение графика	729
Глава 23. Смарт-теги	735
Создание смарт-тега на основе XML-файла.....	735
Применение регулярных выражений.....	737
Создание смарт-тега на основе DLL в VB 6.....	740
Пример создания смарт-тега на основе DLL.....	741
ПРИЛОЖЕНИЯ	749
Приложение 1. Основы программирования на VBA	751
Типы данных.....	751
Переменные	751
Время жизни и область видимости переменных	752
Определение переменных с помощью директивы <i>Option Explicit</i>	753
Ключевое слово <i>Null</i>	753
Комментарии	753
Перенос строки кода.....	753
Строки	754
Даты	754
Массивы	755
Динамические массивы	756
Определение границ индексов массива.....	757
Константы	758
Перечисляемый тип	758
Тип данных, определенный пользователем	759
Математические операции	760
Операции отношения	760
Логические операции.....	761
Директива <i>Option Compare</i>	761
Математические функции	762
Генерация последовательности случайных чисел.....	763
Число π	764
Функции проверки типов.....	764
Преобразование строки в число.....	765
Преобразование числа в строку.....	765
Другие функции преобразования типов.....	766

Форматирование числа функцией <i>Format</i>	766
Пользовательские форматы.....	769
Форматирование чисел функцией <i>FormatNumber</i>	770
Форматирование процентов функцией <i>FormatPercent</i>	771
Задание денежного формата функцией <i>FormatCurrency</i>	771
Форматирование даты и времени функцией <i>FormatDateTime</i>	772
Встроенные диалоговые окна	772
Окно ввода	772
Проверка вводимых данных.....	774
Окно сообщения.....	775
Определение нажатой кнопки в окне сообщений.....	778
Оператор присваивания.....	779
Присваивание переменной ссылки на объект	779
Оператор <i>With</i>	780
Операторы управления	780
Оператор условного перехода	780
Отказ от использования оператора условного перехода.....	781
Оператор выбора	782
Оператор <i>For Next</i>	783
Оператор <i>For Each</i>	784
Оператор <i>While</i>	784
Оператор <i>Do</i>	785
Оператор безусловного перехода <i>GoTo</i>	786
Процедура	786
Передача параметров по ссылке и значению.....	789
Рекурсивные процедуры	790
Фракталы	791
Меры предосторожности при использовании рекурсии.....	795
Приложение 2. VBA и игры	797
Игра "Ход конем"	797
Игра "Крестики-нолики"	805
Предметный указатель	813

Глава 4



Работа со строками, временем и датами

Функции обработки строк

В VBA имеются следующие функции обработки строковых выражений, которые позволяют произвести широкий спектр операций от нахождения длины строки, до замены в строке подстрок:

Asc	Chr	LCase	UCase
Left	Right	Mid	Len
LTrim	RTrim	Trim	Space
Split	Join	String	StrComp
StrReverse	InStr	InStrRev	Replace
StrConv			

Все эти функции с примерами подробно опишем в следующих разделах.

Нахождение ASCII-кода по литералу и литерала по ASCII-коду

Функция `Asc` возвращает ASCII-код первой буквы строки. Например, две следующие инструкции выведут значение 72 в окно **Immediate**.

```
Debug.Print Asc("Hello")  
Debug.Print Asc("H")
```

Функция `Chr` преобразует ASCII-код в строку.

```
Chr(Charcode)
```

Здесь *Charcode* принимает значения от 0 до 255. Значения от 0 до 31 соответствуют так называемым *управляющим кодам*. Например, `Chr(13)` возвращает символ перехода на следующую строку, а `Chr(97)` — символ "a".

Для обозначения некоторых наиболее часто употребляемых клавиш существуют *встроенные константы* VBA, например, для клавиши <Backspace> — vbBack, клавиши <Tab> — vbTab, клавиши <Enter> — vbCrLf.

В следующем примере в окне в первой строчке будет отображаться Visual Basic, а во второй — for Applications:

```
MsgBox "Visual Basic" & vbCrLf & "for Applications"
```

Преобразование строки к нижнему или верхнему регистру

Функции LCase и UCase преобразуют строку к нижнему и верхнему регистру. Например, следующие инструкции отобразят HELLO и hello.

```
Debug.Print LCase("Hello")
```

```
Debug.Print UCase("Hello")
```

Преобразование первых букв слов в прописные

Функция StrConv может произвести более тонкие преобразования строки, чем функции LCase и UCase. Если значение ее второго параметра равно vbUpperCase или vbLowerCase, то производится перевод всех букв в верхний или нижний регистр, а если равно vbProperCase, то в верхний регистр переводятся первые буквы каждого слова строки.

```
Dim strName As String
```

```
strName = "андрей гарнаев"
```

```
strName = StrConv(strName, vbProperCase)
```

```
MsgBox strName ' Отобразится Андрей Гарнаев
```

Возвращение подстроки по указанному местоположению

Функции Left и Right возвращают подстроки, состоящие из заданного числа соответственно первых и последних символов данной строки. Функция Mid возвращает подстроку строки, содержащую указанное число символов, начиная со специфицированной позиции. Например,

```
Debug.Print Left("Hello, World", 5) ' Отобразится Hello
```

```
Debug.Print Right("Hello, World", 5) ' Отобразится World
```

```
Debug.Print Mid("Hello, World", 8, 2)
```

```
' Отобразится Wo, т. к. это два символа, начиная с 8-й позиции
```

Как узнать, содержит ли строка число?

В качестве демонстрационного примера использования функции `Mid` приведем функцию, которая определяет, содержит ли данная строка число.

```
Function ContainsNumbers(strString As String) As Boolean
    Dim i As Integer
    For i = 1 To Len(strString)
        If IsNumeric(Mid(strString, i, 1)) Then
            ContainsNumbers = True
            Exit For
        End If
    Next i
End Function
```

Определение длины строки

Функция `Len` возвращает длину строки. Например, следующий код выводит данную строку посимвольно.

```
Dim i As Integer
Dim s As String
s = "Hello, World"
For i = 1 To Len(s)
    Debug.Print Mid(s, i, 1)
Next
```

Копия строки без начальных и конечных пробелов

Функции `LTrim`, `RTrim` возвращают копию строки без пробелов соответственно в ее начале и конце, а функция `Trim` — без пробелов как в начале, так и в конце строки. Например, в следующем коде переменной `sl` будет присвоено значение "Hello, World", переменной `sr` — значение " Hello, World", а переменной `st` — значение "Hello, World".

```
Dim s As String, sl As String, sr As String, st As String
s = " Hello, World "
sl = LTrim(s)
sr = RTrim(s)
st = Trim(s)
```

Строка, состоящая из указанного числа пробелов

Функция `Space` возвращает строку, состоящую из указанного числа пробелов. Например, в следующем коде (листинг 4.1) за счет последовательного добавления перед данной строкой другой строки, состоящей каждый раз из меньшего числа пробелов, создается эффект бегущей строки.

Листинг 4.1. Бегущая строка

```
Private n As Integer

Sub LetsGo()
    n = 10
    RunString
End Sub

Sub RunString()
    If n >= 0 Then
        Range("A1").Value = Space(n) & "Hello, World!"
        n = n - 1
        Application.OnTime Now + TimeValue("00:00:01"), "RunString"
    End If
End Sub
```

Строка, состоящая из указанного числа повторяющихся символов

Функция `String` возвращает строку, состоящую из указанного числа повторений одного и того же символа.

`String(Number, Character)`

Здесь:

- *Number* — число повторов символа;
- *Character* — повторяемый символ.

Например, функция `FormatString` возвращает строку, указанную в качестве ее параметра `sourceString` и состоящую из столько символов, сколько указано в качестве значения ее параметра `size`. А именно, если значение параметра `size` меньше длины данной строки, то возвращаемая строка является

усечением исходной. Если значение этого параметра больше длины исходной строки, то возвращается строка, состоящая из данной, и присоединенных к ней справа недостающих до специфицированной длины пробелов (листинг 4.2).

Листинг 4.2. Строка, состоящая из указанного числа повторяющихся символов

```
Function FormatString(sourceString As String, size As Integer) As String
    Dim l As Integer
    l = Len(sourceString)
    If l < size Then
        FormatString = sourceString + String(size - l, " ")
    Else
        FormatString = Left(sourceString, size)
    End If
End Function
```

Преобразование строки в массив

Функция `Split` преобразует строку в массив.

```
Split(Expression, [Delimiter], [Limit], [Compare])
```

Здесь:

- *Expression* — преобразуемое строковое выражение;
- *Delimiter* — разделитель между элементами строкового выражения, из которых создается массив. По умолчанию используется пробел;
- *Limit* — максимальное число элементов в массиве. Если значение параметра равно -1 , то нет ограничения на размер массива;
- *Compare* — целое число, указывающее критерий отбора строк.

Например, следующий код разобьет строку на составляющие ее слова и предлоги.

```
Dim s As String
Dim m As Variant
Dim i As Integer
s = "Алиса в стране чудес"
m = Split(s)
For i = LBound(m) To UBound(m)
    Debug.Print m(i)
Next
```

Преобразование массива в строку

Функция `Join` преобразует массив в строку. При этом в качестве значения необязательного параметра можно указать разделитель между элементами строкового выражения, из которых создается массив. По умолчанию используется пробел. Например,

```
Dim m As Variant
Dim s As String
m = Array("Вера", "Надежда", "Любовь")
s = Join(m, ", ")
```

В результате переменной `s` будет присвоено значение "Вера, Надежда, Любовь".

Сравнение строк

Функция `StrComp` возвращает результат сравнения двух строк.

```
StrComp(String1, String2 [, Compare])
```

Здесь:

- `String1` и `String2` — два любых строковых выражения;
- `Compare` — указывает способ сравнения строк. Допустимые значения: `vbBinaryCompare` или 0 (двоичное сравнение), `vbTextCompare` или 1 (посимвольное сравнение без учета регистра).

Функция `StrComp` возвращает следующие значения:

- если `String1` меньше чем `String2`, то -1;
- если `String1` равняется `String2`, то 0;
- если `String1` больше чем `String2`, то 1.

Например,

```
Debug.Print StrComp("Hello", "hello", vbBinaryCompare)
' Возвращает -1
Debug.Print StrComp("Hello", "Hello", vbTextCompare)
' Возвращает 0
Debug.Print StrComp("Hello", "He", vbBinaryCompare)
' Возвращает 1
Debug.Print StrComp("Hello", "Hello, World", vbTextCompare)
' Возвращает -1
```

Создание зеркальной строки по отношению к данной

Функция `StrReverse` возвращает строку, состоящую из тех же символов, что и данная, но расположенных в обратном порядке. Например:

```
Debug.Print StrReverse("Hello") ' Возвращает olleH
```

Нахождение вхождения в строку подстроки

Функции `InStr`, `InStrRev` возвращают позиции первого вхождения подстроки в данную строку, начиная соответственно с ее начала и с конца. Если вхождения нет, то возвращается значение 0.

```
InStr([Start, ] StringCheck, StringMatch[, Compare])  
InStrRev(StringCheck, StringMatch, [Start], [Compare])
```

где

- *Start* — числовое выражение, задающее позицию, с которой начинается каждый поиск. Если этот параметр опущен, то поиск начинается с первого символа строки;
- *StringCheck* — строковое выражение, в котором выполняется поиск;
- *StringMatch* — искомое строковое выражение;
- *Compare* — указывает способ сравнения строк. Допустимые значения: `vbBinaryCompare` или 0 (для двоичного сравнения), `vbTextCompare` или 1 (посимвольное сравнение без учета регистра).

Например:

```
Debug.Print InStr("Hello", "lo") ' Возвращает 4  
Debug.Print InStr("Hello", "wo") ' Возвращает 0
```

Замена в строке подстроки

Функция `Replace` заменяет в строке подстроку.

```
Replace(Expression, Find, Replacewith[, Start[, Count [, Compare]])]
```

Здесь:

- *Expression* — строка, в которой заменяется подстрока;
- *Find* — заменяемая подстрока;
- *Replacewith* — подстрока, на которую заменяется подстрока, указанная в качестве значения параметра *Find*;

- ❑ *Start* — позиция в строке, с которой ищется подстрока, указанная в качестве значения параметра *Find*. Если этот параметр опущен, то поиск производится с первой позиции;
- ❑ *Count* — указывает, сколько найденных подстрок надо заменить. Если параметр опущен, то будут произведены все замены;
- ❑ *Compare* — критерий сравнения при поиске подстроки. Допустимые значения: 0 (для двоичного сравнения), 1 (посимвольное сравнение без учета регистра).

Например:

```
Dim s As String, r As String
s = "Как хороши, как свежи были розы"
r = Replace(s, "как", "так", , , 1)
```

В результате получится строка "так хороши, так свежи были розы".

Проверка, является ли строка электронным адресом

В качестве примера работы со строковыми функциями приведем функцию (листинг 4.3), которая проверяет, является ли данная строка электронным адресом. Если она таковым не является, то функция возвращает значение `False`, а если является, то `True`.

Листинг 4.3. Проверка, является ли строка электронным адресом

```
Function IsEmail(email As String) As Boolean
    IsEmail = True
    Dim names As Variant, name As Variant
    Dim i As Integer, c As String
    names = Split(email, "@")
    If UBound(names) <> 1 Then
        IsEmail = False
        Exit Function
    End If
    For Each name In names
        If Len(name) <= 0 Then
            IsEmail = False
            Exit Function
        End If
        For i = 1 To Len(name)
```

```

    c = LCase(Mid(name, i, 1))
    If InStr("abcdefghijklmnopqrstuvwxyz_-.", c) <= 0 _
        And Not IsNumeric(c) Then
        IsEmail = False
        Exit Function
    End If
Next
If Left(name, 1) = "." Or Right(name, 1) = "." Then
    IsEmail = False
    Exit Function
End If
Next
If InStr(names(1), ".") <= 0 Then
    IsEmail = False
    Exit Function
End If
i = Len(names(1)) - InStrRev(names(1), ".")
If i <> 2 And i <> 3 Then
    IsEmail = False
    Exit Function
End If
If InStr(email, "..") > 0 Then
    IsEmail = False
End If
End Function

```

Функции даты и времени

В VBA имеются следующие функции времени и даты, которые позволяют произвести широкий спектр операций от определения текущей даты до сложения указанных дат:

Date	DateAdd	DateDiff	DatePart
DateSerial	Day	Hour	Minute
Month	Now	Second	Time
Timer	TimeSerial	TimeValue	Weekday
Year			

В последующих разделах на примерах рассмотрены все эти функции.

Определение текущей системной даты

Функция `Date` возвращает значение типа `Variant (Date)`, содержащее текущую системную дату. Например, данная строка писалась 12 августа 2005 г., поэтому инструкция

```
Debug.Print Date
```

выводит значение 12.08.2005.

Определение текущего системного времени

Функция `Time` возвращает значение типа `Variant (Date)`, содержащее текущее время по системным часам компьютера. Например, данная строка писалась в 20 часов 46 минут и 14 секунд 12 августа 2005 г., поэтому инструкция

```
Debug.Print Time
```

выводит значение 20:46:14.

Определение текущей даты и системного времени

Функция `Now` возвращает значение типа `Variant (Date)`, содержащее текущую дату и время по системному календарю и часам компьютера. Например, данная строка писалась в 20 часов 49 минут и 24 секунды 12 августа 2005 г., поэтому инструкция

```
Debug.Print Now
```

выводит значение 12.08.2005 20:49:24.

Извлечение из времени часового, минутного и секундного компонентов

Функции `Hour`, `Minute`, `Second` возвращают значения типа `Variant (Integer)`, являющиеся целыми числами и представляющие часы, минуты и секунды в значении времени. В следующем примере переменным `h`, `m` и `s` присваиваются значения 16, 35 и 17.

```
Dim dt As Date
```

```
Dim h, m, s As Integer
```

```
dt = #4:35:17 PM#
```

```
h = Hour(dt)
```

```
m = Minute(dt)
```

```
s = Second(dt)
```