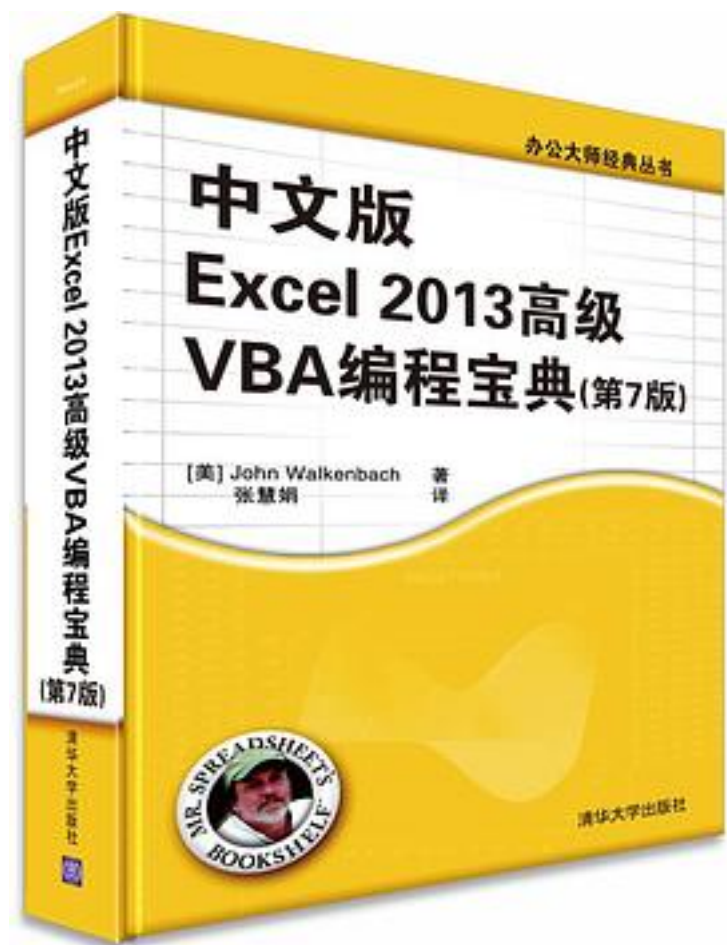


中文版Excel 2013高级VBA 编程宝典(第7 版)



[中文版Excel 2013高级VBA 编程宝典\(第7 版\) 下载链接1](#)

著者:(美) 沃肯巴赫(Walkenbach, J.) 著

出版者:清华大学出版社

出版时间:2014-9-1

装帧:平装

isbn:9787302376712

通过学习VBA编程技术来扩展Excel 2013

如果你出于工作需要或个人爱好，想了解如何开发程序来增强Excel并自动完成计算任务，那么“电子表格先生”JohnWalkenbach无疑是你最好的老师。John撰写的《中文版Excel 2013高级VBA编程宝典(第7版)》全面介绍VBA，精确讲解如何创建自定义的对话框和用户窗体，还呈现诸多高级编程技术，如开发实用程序以及使用VBA处理透视表和图表等。

通过学习本书，你还将能创建面向用户的应用程序、修改快捷菜单并为自己的应用程序提供在线帮助。如果你已初步掌握Excel的用法，并力求提升Excel技能水准，那么本书将是你的理想选择。

作者介绍:

John

Walkenbach是全球顶尖的Excel权威专家，迄今已撰写50多本书籍，包括畅销多年的《Excel宝典》系列、《Excel公式》系列、《Excel高级VBA编程宝典》系列等。此外，John已在PCWorld、InfoWorld和Windows等刊物上发表300多篇技术文章。由他开发的PowerUtility Pak工具箱屡获大奖。John维护着电子表格网站www.spreadsheetpage.com，其中展现了大量Excel信息和专家意见。

目录: 第 I 部分 Excel基础知识

第1章 Excel概述 3

1.1 关于Excel 3

1.2 关于对象的思考 3

1.3 工作簿 4

1.3.1 工作表 4

1.3.2 图表工作表 5

1.3.3 XLM宏工作表 6

1.3.4 Excel 5/95对话框编辑表 7

1.4 Excel的用户界面 7

1.4.1 关于功能区 8

1.4.2 快捷菜单和浮动工具栏 13

1.4.3 对话框 14

1.4.4 任务窗格 15

1.4.5 键盘快捷键 16

1.5 数据录入 16

1.6 公式、函数和名称 16

1.7 选择对象 19

1.8 格式 19

1.9 保护选项 20

1.9.1 保护公式以防被重写 20

1.9.2 保护工作簿的结构 21

1.9.3 使用密码来保护工作簿 21

1.9.4 使用密码来保护VBA代码 22

1.10 图表 23

1.11 形状和SmartArt 23

1.12 数据库访问 24

1.12.1 工作表数据库 24

1.12.2 外部数据库 25

1.13 Internet特性 25

1.14 分析工具 26

1.15 加载项 27

1.16 宏和编程	27
1.17 文件格式	28
1.18 Excel的帮助系统	28
第2章 公式的使用技巧	29
2.1 公式概述	29
2.2 计算公式	30
2.3 单元格引用和单元格区域引用	31
2.3.1 为什么使用的不是相对引用	31
2.3.2 R1C1表示法	32
2.3.3 引用其他工作表或工作簿	32
2.4 使用名称	34
2.4.1 命名单元格和单元格区域	34
2.4.2 将名称应用于现有的引用中	35
2.4.3 交叉名称	36
2.4.4 命名列和行	36
2.4.5 名称的作用范围	36
2.4.6 命名常量	37
2.4.7 命名公式	38
2.4.8 命名对象	40
2.5 公式错误	40
2.6 Excel审核工具	41
2.6.1 识别特定类型的单元格	41
2.6.2 查看公式	42
2.6.3 追踪单元格关系	44
2.6.4 追踪错误值	45
2.6.5 修正循环引用错误	46
2.6.6 使用后台错误检查功能	46
2.6.7 使用Excel的公式求值	47
2.7 数组公式	48
2.7.1 数组公式的一个示例	49
2.7.2 数组公式日历	50
2.7.3 数组公式的优缺点	51
2.8 计数和求和技巧	51
2.8.1 计数公式的示例	52
2.8.2 求和公式示例	52
2.8.3 其他计数工具	53
2.9 查找公式	53
2.10 使用日期和时间	55
2.10.1 输入日期和时间	56
2.10.2 使用1900年之前的日期	56
2.11 创建大公式	57
第3章 理解Excel的文件	61
3.1 启动Excel	61
3.2 文件类型	62
3.2.1 Excel文件格式	62
3.2.2 文本文件格式	63
3.2.3 数据库文件格式	64
3.2.4 其他文件格式	64
3.3 Excel文件兼容性	65
3.4 受保护的视图	66
3.5 使用自动恢复	67
3.5.1 恢复当前工作簿的各个版本	67
3.5.2 恢复未保存的工作	67
3.5.3 配置自动恢复	67

3.6 模板文件的处理	68
3.6.1 查看模板	68
3.6.2 创建模板	68
3.6.3 创建工作簿模板	69
3.7 Excel文件的内部情况	70
3.7.1 仔细分析一个文件	70
3.7.2 为什么文件格式很重要	73
3.8 OfficeUI文件	74
3.9 XLB文件	75
3.10 加载项文件	75
3.11 Excel在注册表中的设置	76
3.11.1 关于注册表	76
3.11.2 Excel的设置	77
第4章 电子表格应用程序开发的基础知识	79
4.1 什么是电子表格应用程序	79
4.2 电子表格应用程序的开发步骤	80
4.3 确定用户需求	80
4.4 规划满足用户需求的应用程序	81
4.5 确定最合适的用户界面	83
4.5.1 创建自定义功能区	83
4.5.2 创建自定义快捷菜单	83
4.5.3 创建快捷键	84
4.5.4 创建自定义对话框	84
4.5.5 在工作表上使用ActiveX控件	85
4.5.6 执行开发的成果	87
4.6 关心终端用户	87
4.6.1 测试应用程序	87
4.6.2 尽量完善应用程序的安全性	89
4.6.3 让应用程序变得时尚且直观	90
4.6.4 创建用户帮助系统	91
4.6.5 将开发工作进行归档	92
4.6.6 将电子表格应用程序分发给用户	92
4.6.7 在需要的时候更新电子表格应用程序	92
4.7 其他开发问题	93
4.7.1 用户安装的Excel版本	93
4.7.2 语言问题	93
4.7.3 系统速度	93
4.7.4 视频模式	94
第II部分 理解VBA	
第5章 VBA概述	97
5.1 BASIC的一些基本背景	97
5.2 关于VBA	98
5.2.1 对象模型	98
5.2.2 VBA与XLM的对比	98
5.3 VBA的基础知识	99
5.4 Visual Basic编辑器概述	101
5.4.1 显示Excel的“开发工具”选项卡	102
5.4.2 激活VBE	102
5.4.3 VBE窗口	103
5.5 使用“工程资源管理器”窗口	103
5.5.1 添加新的VBA模块	105
5.5.2 移除VBA模块	105
5.5.3 导出和导入对象	105
5.6 使用“代码”窗口	105

5.6.1 窗口的最小化和最大化	106
5.6.2 VBA代码的存储	106
5.6.3 VBA代码的输入	107
5.7 VBE环境的定制	112
5.7.1 使用“编辑器”选项卡	113
5.7.2 使用“编辑器格式”选项卡	116
5.7.3 使用“通用”选项卡	116
5.7.4 使用“可连接的”选项卡	118
5.8 宏录制器	118
5.8.1 宏录制器实际记录哪些内容	119
5.8.2 相对模式还是绝对模式	120
5.8.3 选项的录制	122
5.8.4 清理已录制的宏	123
5.9 关于对象和集合	125
5.9.1 对象层次结构	125
5.9.2 关于集合	125
5.9.3 对象的引用	126
5.10 属性和方法	127
5.10.1 对象的属性	127
5.10.2 对象的方法	128
5.11 Comment对象示例	129
5.11.1 查看有关Comment对象的帮助	129
5.11.2 Comment对象的属性	131
5.11.3 Comment对象的方法	131
5.11.4 Comments集合	132
5.11.5 关于Comment属性	133
5.11.6 Comment对象中的对象	133
5.11.7 确定单元格中是否含有Comment对象	134
5.11.8 添加新的Comment对象	135
5.12 一些有用的应用程序属性	135
5.13 Range对象的使用	137
5.13.1 Range属性	137
5.13.2 Cells属性	139
5.13.3 Offset属性	141
5.14 关于对象的更多信息	142
5.14.1 需要牢记的基本概念	142
5.14.2 学习有关对象和属性的更多信息	143
第6章 VBA编程基础	147
6.1 VBA语言元素概览	147
6.2 注释	149
6.3 变量、数据类型和常量	150
6.3.1 数据类型的定义	151
6.3.2 声明变量	153
6.3.3 变量的作用域	155
6.3.4 常量的使用	159
6.3.5 字符串的使用	160
6.3.6 日期的使用	161
6.4 赋值语句	162
6.5 数组	164
6.5.1 数组的声明	164
6.5.2 多维数组的声明	164
6.5.3 动态数组的声明	165
6.6 对象变量	165
6.7 用户自定义数据类型	166

- 6.8 内置函数 167
- 6.9 对象和集合的处理 170
 - 6.9.1 With-End With结构 170
 - 6.9.2 For Each-Next结构 171
- 6.10 代码执行的控制 173
 - 6.10.1 GoTo语句 173
 - 6.10.2 If-Then结构 174
 - 6.10.3 Select Case结构 177
 - 6.10.4 指令块的循环 181
- 第7章 VBA的Sub过程 189
 - 7.1 关于过程 189
 - 7.1.1 Sub过程的声明 190
 - 7.1.2 过程的作用域 190
 - 7.2 执行Sub过程 191
 - 7.2.1 通过“运行子过程/用户窗体”命令执行过程 192
 - 7.2.2 从“宏”对话框执行过程 192
 - 7.2.3 用Ctrl+快捷键组合执行过程 193
 - 7.2.4 从功能区执行过程 194
 - 7.2.5 从自定义快捷菜单中执行过程 194
 - 7.2.6 从另一个过程中执行过程 195
 - 7.2.7 通过单击对象执行过程 198
 - 7.2.8 在事件发生时执行过程 199
 - 7.2.9 从“立即窗口”执行过程 200
 - 7.3 向过程中传递参数 201
 - 7.4 错误处理技术 204
 - 7.4.1 捕获错误 204
 - 7.4.2 错误处理示例 205
 - 7.5 使用Sub过程的实际示例 208
 - 7.5.1 目标 208
 - 7.5.2 工程需求 208
 - 7.5.3 已经了解的信息 209
 - 7.5.4 解决方法 209
 - 7.5.5 需要了解哪些信息 210
 - 7.5.6 初步的录制工作 210
 - 7.5.7 初始设置 211
 - 7.5.8 代码的编写 212
 - 7.5.9 排序过程的编写 213
 - 7.5.10 更多的测试 217
 - 7.5.11 修复问题 217
 - 7.5.12 实用程序的可用性 221
 - 7.5.13 对工程进行评估 222
- 第8章 创建Function过程 223
 - 8.1 Sub过程与Function过程的比较 223
 - 8.2 为什么创建自定义的函数 224
 - 8.3 介绍性的函数示例 224
 - 8.3.1 在工作表中使用函数 225
 - 8.3.2 在VBA过程中使用函数 226
 - 8.3.3 分析自定义函数 226
 - 8.4 Function过程 228
 - 8.4.1 函数的作用域 229
 - 8.4.2 执行Function过程 229
 - 8.5 Function过程的参数 232
 - 8.6 函数示例 233
 - 8.6.1 无参数的函数 234

- 8.6.2 带有一个参数的函数 236
- 8.6.3 带有两个参数的函数 238
- 8.6.4 使用数组作为参数的函数 239
- 8.6.5 带有可选参数的函数 240
- 8.6.6 返回VBA数组的函数 242
- 8.6.7 返回错误值的函数 245
- 8.6.8 带有不定数量参数的函数 246
- 8.7 模拟Excel的SUM函数 247
- 8.8 扩展的日期函数 250
- 8.9 函数的调试 252
- 8.10 使用“插入函数”对话框 253
 - 8.10.1 使用MacroOptions方法 254
 - 8.10.2 指定函数类别 255
 - 8.10.3 手动添加函数说明 256
- 8.11 使用加载项存储自定义的函数 257
- 8.12 使用Windows API 258
 - 8.12.1 Windows API示例 259
 - 8.12.2 确定Windows目录 259
 - 8.12.3 检测Shift键 260
 - 8.12.4 了解更多有关API函数的信息 261
- 第9章 VBA编程示例和技巧 263
 - 9.1 通过示例学习 263
 - 9.2 处理单元格区域 264
 - 9.2.1 复制单元格区域 264
 - 9.2.2 移动单元格区域 265
 - 9.2.3 复制大小可变的单元格区域 265
 - 9.2.4 选中或者识别各种类型的单元格区域 267
 - 9.2.5 调整单元格区域大小 269
 - 9.2.6 提示输入单元格中的值 269
 - 9.2.7 在下一个空单元格中输入一个值 271
 - 9.2.8 暂停宏的运行以便获得用户选中的单元格区域 272
 - 9.2.9 计算选中的单元格的数目 274
 - 9.2.10 确定选中的单元格区域的类型 274
 - 9.2.11 有效地循环遍历选中的单元格区域 276
 - 9.2.12 删除所有空行 279
 - 9.2.13 任意次数地复制行 280
 - 9.2.14 确定单元格区域是否包含在另一个单元格区域内 281
 - 9.2.15 确定单元格的数据类型 281
 - 9.2.16 读写单元格区域 283
 - 9.2.17 在单元格区域中写入值的更好方法 284
 - 9.2.18 传递一维数组中的内容 286
 - 9.2.19 将单元格区域传递给Variant类型的数组 286
 - 9.2.20 按数值选择单元格 287
 - 9.2.21 复制非连续的单元格区域 288
 - 9.3 处理工作簿和工作表 290
 - 9.3.1 保存所有工作簿 290
 - 9.3.2 保存和关闭所有工作簿 291
 - 9.3.3 隐藏除选区之外的区域 291
 - 9.3.4 创建超链接内容表 292
 - 9.3.5 同步工作表 293
 - 9.4 VBA技巧 294
 - 9.4.1 切换布尔类型的属性值 294
 - 9.4.2 显示日期和时间 295
 - 9.4.3 显示友好时间 296

- 9.4.4 获得字体列表 298
- 9.4.5 对数组进行排序 299
- 9.4.6 处理一系列文件 301
- 9.5 使用在代码中的一些有用的函数 302
 - 9.5.1 FileExists函数 302
 - 9.5.2 FileNameOnly函数 303
 - 9.5.3 PathExists函数 303
 - 9.5.4 RangeNameExists函数 303
 - 9.5.5 SheetExists函数 305
 - 9.5.6 WorkbookIsOpen函数 305
 - 9.5.7 检索已经关闭的工作簿中的值 305
- 9.6 一些有用的工作表函数 307
 - 9.6.1 返回单元格的格式信息 307
 - 9.6.2 会说话的工作表 308
 - 9.6.3 显示在保存或打印文件时的时间 309
 - 9.6.4 理解对象的父对象 310
 - 9.6.5 计算介于两个值之间的单元格数目 311
 - 9.6.6 确定行或列中最后一个非空的单元格 311
 - 9.6.7 字符串与模式匹配 313
 - 9.6.8 从字符串中提取第n个元素 314
 - 9.6.9 拼写出数字 315
 - 9.6.10 多功能的函数 315
 - 9.6.11 SHEETOFFSET函数 316
 - 9.6.12 返回所有工作表中数据的最大值 317
 - 9.6.13 返回没有重复的随机整数元素的数组 318
 - 9.6.14 随机化单元格区域 319
 - 9.6.15 对单元格区域进行排序 321
- 9.7 Windows API调用 322
 - 9.7.1 确定文件的关联性 323
 - 9.7.2 确定磁盘驱动器信息 324
 - 9.7.3 确定默认打印机的信息 324
 - 9.7.4 确定视频显示器的信息 325
 - 9.7.5 读写注册表 327
- 第Ⅲ部分 操作用户窗体
- 第10章 多种自定义对话框的方法 333
 - 10.1 创建用户窗体之前需要了解的内容 333
 - 10.2 使用输入框 333
 - 10.2.1 VBA的InputBox函数 334
 - 10.2.2 Excel的InputBox方法 336
 - 10.3 VBA的MsgBox函数 339
 - 10.4 Excel的GetOpenFilename方法 343
 - 10.5 Excel的GetSaveAsFilename方法 346
 - 10.6 提示输入目录名称 347
 - 10.7 显示Excel的内置对话框 347
 - 10.8 显示数据记录单 350
 - 10.8.1 使得数据记录单变得可以访问 350
 - 10.8.2 通过使用VBA来显示数据记录单 351
- 第11章 用户窗体概述 353
 - 11.1 Excel如何处理自定义对话框 353
 - 11.2 插入新的用户窗体 354
 - 11.3 向用户窗体中添加控件 354
 - 11.4 “工具箱”中的控件 355
 - 11.4.1 复选框 356
 - 11.4.2 组合框 356

- 11.4.3 命令按钮 356
- 11.4.4 框架 356
- 11.4.5 图像 357
- 11.4.6 标签 357
- 11.4.7 列表框 357
- 11.4.8 多页 357
- 11.4.9 选项按钮 357
- 11.4.10 RefEdit 357
- 11.4.11 滚动条 358
- 11.4.12 数值调节钮 358
- 11.4.13 TabStrip 358
- 11.4.14 文本框 358
- 11.4.15 切换按钮 358
- 11.5 调整用户窗体的控件 359
- 11.6 调整控件的属性 361
 - 11.6.1 使用“属性”窗口 361
 - 11.6.2 共同属性 362
 - 11.6.3 满足键盘用户的需求 363
- 11.7 显示用户窗体 364
 - 11.7.1 调整显示位置 365
 - 11.7.2 显示无模式的用户窗体 365
 - 11.7.3 显示基于变量的用户窗体 366
 - 11.7.4 加载用户窗体 366
 - 11.7.5 关于事件处理程序 366
- 11.8 关闭用户窗体 366
- 11.9 创建用户窗体的示例 368
 - 11.9.1 创建用户窗体 368
 - 11.9.2 编写代码显示对话框 370
 - 11.9.3 测试对话框 371
 - 11.9.4 添加事件处理程序 372
 - 11.9.5 验证数据的有效性 373
 - 11.9.6 完成的对话框 374
- 11.10 理解用户窗体的事件 374
 - 11.10.1 了解事件 374
 - 11.10.2 用户窗体的事件 375
 - 11.10.3 数值调节钮的事件 376
 - 11.10.4 数值调节钮与文本框配套使用 377
- 11.11 引用用户窗体的控件 379
- 11.12 自定义“工具箱” 381
 - 11.12.1 在“工具箱”中添加新页 381
 - 11.12.2 自定义或组合控件 381
 - 11.12.3 添加新的ActiveX控件 382
- 11.13 创建用户窗体的模板 383
- 11.14 用户窗体检验表 384
- 第12章 用户窗体示例 385
 - 12.1 创建用户窗体式菜单 385
 - 12.1.1 在用户窗体中使用命令按钮 385
 - 12.1.2 在用户窗体中使用列表框 386
 - 12.2 从用户窗体选中单元格区域 387
 - 12.3 创建欢迎界面 389
 - 12.4 禁用用户窗体的关闭按钮 391
 - 12.5 改变用户窗体的大小 392
 - 12.6 从用户窗体中缩放和滚动工作表 393
 - 12.7 列表框技巧 395

- 12.7.1 向列表框控件中添加条目 396
- 12.7.2 确定选中的条目 400
- 12.7.3 确定选中的列表框中的多个条目 401
- 12.7.4 单个列表框中的多个列表 402
- 12.7.5 列表框条目的转移 402
- 12.7.6 在列表框中移动条目 404
- 12.7.7 使用多列的列表框控件 405
- 12.7.8 使用列表框选中工作表中的行 407
- 12.7.9 使用列表框激活工作表 410
- 12.8 在用户窗体中使用多页控件 412
- 12.9 使用外部控件 413
- 12.10 使标签动画化 416
- 第13章 用户窗体的高级技巧 419
 - 13.1 无模式对话框 419
 - 13.2 显示进度条 423
 - 13.2.1 创建独立的进度条 424
 - 13.2.2 使用多页控件显示进度条 427
 - 13.2.3 在不使用多页控件的情况下显示进度条 429
 - 13.3 创建向导 431
 - 13.3.1 为向导设置多页控件 431
 - 13.3.2 在向导用户窗体中添加按钮 432
 - 13.3.3 编写向导按钮的程序 432
 - 13.3.4 编写向导中的相关代码 434
 - 13.3.5 使用向导执行任务 435
 - 13.4 模仿MsgBox函数 436
 - 13.4.1 模仿MsgBox函数：MyMsgBox函数的代码 437
 - 13.4.2 MyMsgBox函数的工作原理 438
 - 13.4.3 使用MyMsgBox函数 440
 - 13.5 带有可移动控件的用户窗体 440
 - 13.6 没有标题栏的用户窗体 441
 - 13.7 使用用户窗体模拟工具栏 443
 - 13.8 使用用户窗体来模仿任务面板 445
 - 13.9 可调整大小的用户窗体 446
 - 13.10 用一个事件处理程序处理多个用户窗体控件 450
 - 13.11 在用户窗体中选择颜色 452
 - 13.12 在用户窗体中显示图表 454
 - 13.12.1 将图表保存为GIF文件 455
 - 13.12.2 更改图像控件的Picture属性 455
 - 13.13 使用用户窗体半透明 455
 - 13.14 增强型数据记录单 457
 - 13.14.1 关于Enhanced Data Form加载项 459
 - 13.14.2 安装Enhanced Data Form加载项 459
 - 13.15 用户窗体上的数字推盘 460
 - 13.16 用户窗体上的电动扑克 461
- 第IV部分 高级编程技术
- 第14章 用VBA开发Excel实用程序 465
 - 14.1 关于Excel实用程序 465
 - 14.2 使用VBA开发实用程序 466
 - 14.3 如何开发好的实用程序 466
 - 14.4 Text Tools：实用程序剖析 467
 - 14.4.1 Text Tools的背景 468
 - 14.4.2 Text Tools的工程目标 468

- 14.4.3 Text Tools工作簿 468
- 14.4.4 Text Tools实用程序的工作原理 469
- 14.4.5 Text Tools实用程序的用户窗体 470
- 14.4.6 Module1 VBA模块 471
- 14.4.7 UserForm1代码模块 473
- 14.4.8 提高Text Tools实用程序的效率 474
- 14.4.9 保存Text Tools实用程序的设置 475
- 14.4.10 实现撤消 477
- 14.4.11 显示帮助文件 479
- 14.4.12 添加RibbonX代码 480
- 14.4.13 工程验收 481
- 14.4.14 理解Text Tools实用程序 481
- 14.5 获取更多关于Excel实用程序的知识 482
- 第15章 使用数据透视表 483
- 15.1 介绍性数据透视表示例 483
- 15.1.1 创建数据透视表 484
- 15.1.2 检查录制的数据透视表代码 485
- 15.1.3 整理录制的数据透视表代码 486
- 15.2 创建更复杂的数据透视表 488
- 15.2.1 创建数据透视表的代码 489
- 15.2.2 更复杂的数据透视表的工作原理 490
- 15.3 创建多个数据透视表 492
- 15.4 创建转换的数据透视表 495
- 第16章 使用图表 499
- 16.1 关于图表 499
- 16.1.1 图表的位置 499
- 16.1.2 宏录制器和图表 500
- 16.1.3 Chart对象模型 500
- 16.2 创建嵌入式图表 501
- 16.3 在图表工作表上创建图表 503
- 16.4 修改图表 504
- 16.5 使用VBA激活图表 505
- 16.6 图表的移动 506
- 16.7 使用VBA使图表取消激活 506
- 16.8 确定图表是否被激活 507
- 16.9 从ChartObjects或Charts集合中删除图表 508
- 16.10 循环遍历所有图表 509
- 16.11 调整ChartObjects对象的大小并对齐 511
- 16.12 创建大量图表 513
- 16.13 导出图表 515
- 16.14 修改图表中使用的数据 517
- 16.14.1 基于活动单元格修改图表数据 517
- 16.14.2 用VBA确定图表中使用的单元格区域 519
- 16.15 使用VBA在图表上显示任意数据标签 522
- 16.16 在用户窗体中显示图表 526
- 16.17 理解图表事件 528
- 16.17.1 使用图表事件的一个示例 529
- 16.17.2 为嵌入式图表启用事件 531
- 16.17.3 示例：在嵌入式图表上使用Chart事件 532
- 16.18 VBA制图技巧 534
- 16.18.1 在整个页面上打印嵌入式图表 535
- 16.18.2 通过隐藏列来隐藏序列 535
- 16.18.3 创建未链接的图表 536
- 16.18.4 用MouseOver事件显示文本 538

- 16.19 图表动画 540
 - 16.19.1 滚动图表 541
 - 16.19.2 创建一个内摆线图表 543
 - 16.19.3 创建一个“时钟”图表 544
- 16.20 在不使用VBA的情况下创建一个交互式图表 545
 - 16.20.1 获取数据来创建一个交互式图表 546
 - 16.20.2 为交互式工作表创建选项按钮控件 546
 - 16.20.3 为交互式图表创建城市列表 547
 - 16.20.4 创建交互式图表的数据单元格区域 548
 - 16.20.5 创建交互式图表 548
- 16.21 使用迷你图 548
- 第17章 理解Excel的事件 553
 - 17.1 Excel可以监视的事件类型 553
 - 17.1.1 理解事件发生的顺序 554
 - 17.1.2 事件处理程序存放的位置 554
 - 17.1.3 禁用事件 555
 - 17.1.4 输入事件处理代码 556
 - 17.1.5 使用参数的事件处理程序 557
 - 17.2 工作簿级别的事件 559
 - 17.2.1 Open事件 560
 - 17.2.2 Activate事件 561
 - 17.2.3 SheetActivate事件 561
 - 17.2.4 NewSheet事件 561
 - 17.2.5 BeforeSave事件 561
 - 17.2.6 Deactivate事件 562
 - 17.2.7 BeforePrint事件 562
 - 17.2.8 BeforeClose事件 564
 - 17.3 检查工作表事件 565
 - 17.3.1 Change事件 566
 - 17.3.2 监视特定单元格区域的修改 567
 - 17.3.3 SelectionChange事件 571
 - 17.3.4 BeforeDoubleClick事件 572
 - 17.3.5 BeforeRightClick事件 573
 - 17.4 检查图表事件 573
 - 17.5 监视应用程序事件 575
 - 17.5.1 启用应用程序级别的事件 576
 - 17.5.2 确定工作簿何时被打开 576
 - 17.5.3 监视应用程序级别的事件 578
 - 17.6 使用用户窗体事件 579
 - 17.7 访问与对象无关联的事件 580
 - 17.7.1 OnTime事件 580
 - 17.7.2 OnKey事件 582
- 第18章 与其他应用程序的交互 587
 - 18.1 从Excel中启动应用程序 587
 - 18.1.1 使用VBA的Shell函数 587
 - 18.1.2 使用Windows的ShellExecute API函数 590
 - 18.2 用Excel激活应用程序 591
 - 18.2.1 使用AppActivate语句 591
 - 18.2.2 激活一个Microsoft Office应用程序 592
 - 18.3 运行“控制面板”对话框 592
 - 18.4 在Excel中使用自动化 593
 - 18.4.1 通过使用自动化来使用外部对象 594
 - 18.4.2 前期绑定与后期绑定 594
 - 18.4.3 后期绑定的一个示例 597

- 18.4.4 从Excel中控制Word 598
- 18.4.5 从其他应用程序中控制Excel 600
- 18.5 通过Outlook发送个性化的电子邮件 603
- 18.6 从Excel中发送电子邮件附件 606
- 第19章 创建和使用加载项 609
 - 19.1 什么是加载项 609
 - 19.1.1 加载项与标准工作簿的比较 609
 - 19.1.2 创建加载项的原因 610
 - 19.2 理解Excel的加载项管理器 611
 - 19.3 创建加载项 613
 - 19.4 加载项示例 614
 - 19.4.1 为加载项示例添加描述信息 615
 - 19.4.2 创建加载项 615
 - 19.4.3 安装加载项 616
 - 19.4.4 测试加载项 617
 - 19.4.5 发布加载项 617
 - 19.4.6 修改加载项 617
 - 19.5 比较XLAM和XLSM文件 618
 - 19.5.1 XLAM文件中的VBA集合成员 619
 - 19.5.2 XLSM和XLAM文件的可见性 619
 - 19.5.3 XLSM和XLAM文件的工作表和图表工作表 619
 - 19.5.4 访问加载项中的VBA过程 620
 - 19.6 用VBA操作加载项 623
 - 19.6.1 AddIn对象属性 625
 - 19.6.2 作为工作簿访问加载项 628
 - 19.6.3 AddIn对象事件 628
 - 19.7 优化加载项的性能 629
 - 19.8 加载项的特殊问题 630
 - 19.8.1 确保加载项已经安装 630
 - 19.8.2 从加载项中引用其他文件 631
 - 19.8.3 为加载项检测适用的Excel版本 632
- 第V部分 开发应用程序
- 第20章 使用功能区 633
 - 20.1 功能区基础 635
 - 20.2 VBA和功能区 638
 - 20.2.1 访问功能区控件 639
 - 20.2.2 使用功能区 640
 - 20.2.3 激活选项卡 642
 - 20.3 定制功能区 643
 - 20.3.1 RibbonX的一个简单示例 643
 - 20.3.2 功能区的一个简单示例(修改版) 646
 - 20.3.3 RibbonX的另一个示例 650
 - 20.3.4 功能区控件演示 652
 - 20.3.5 dynamicMenu控件示例 659
 - 20.3.6 更多关于功能区定制的内容 661
 - 20.4 创建老式工具栏 662
 - 20.4.1 Excel 2007及后续版本中老式工具栏的局限性 662
 - 20.4.2 创建工具栏的代码 663
- 第21章 使用快捷菜单 667
 - 21.1 命令栏简介 667
 - 21.1.1 命令栏的类型 667
 - 21.1.2 列出快捷菜单 668
 - 21.1.3 引用命令栏 669

- 21.1.4 引用命令栏中的控件 669
- 21.1.5 命令栏控件的属性 671
- 21.1.6 显示所有的快捷菜单项 671
- 21.2 使用VBA定制快捷菜单 673
 - 21.2.1 Excel 2013中的不同之处 674
 - 21.2.2 重置快捷菜单 675
 - 21.2.3 禁用快捷菜单 676
 - 21.2.4 禁用快捷菜单项 677
 - 21.2.5 向“单元格”快捷菜单中添加一个新项 677
 - 21.2.6 向快捷菜单中添加一个子菜单 679
 - 21.2.7 将快捷菜单限制到单个工作簿 681
- 21.3 快捷菜单与事件 682
 - 21.3.1 自动添加和删除菜单 682
 - 21.3.2 禁用或隐藏快捷菜单项 682
 - 21.3.3 创建一个上下文相关的快捷菜单 683
- 第22章 为应用程序提供帮助 687
 - 22.1 Excel应用程序的“帮助” 687
 - 22.2 使用Excel组件的帮助系统 689
 - 22.2.1 为帮助系统使用单元格批注 689
 - 22.2.2 为帮助系统使用文本框 690
 - 22.2.3 使用工作表来显示帮助文本 692
 - 22.2.4 在用户窗体中显示帮助 693
 - 22.3 在Web浏览器中显示“帮助” 696
 - 22.3.1 使用HTML文件 696
 - 22.3.2 使用一个MHTML文件 697
 - 22.4 使用HTML帮助系统 698
 - 22.4.1 使用Help方法来显示HTML帮助 701
 - 22.4.2 将“帮助”文件与应用程序相关联 701
 - 22.4.3 将一个帮助主题与一个VBA函数相关联 702
- 第23章 开发面向用户的应用程序 705
 - 23.1 什么是面向用户的应用程序 705
 - 23.2 Loan Amortization Wizard 705
 - 23.2.1 使用Load Amortization Wizard 706
 - 23.2.2 Load Amortization Wizard的工作簿结构 707
 - 23.2.3 Load Amortization Wizard的工作原理 708
 - 23.2.4 可以增强Load Amortization Wizard的地方 714
 - 23.3 应用程序开发概念 714
- 第VI部分 其他主题
- 第24章 兼容性问题 719
 - 24.1 什么是兼容性 719
 - 24.2 兼容性问题的类型 720
 - 24.3 避免使用新功能 721
 - 24.4 在Mac机器上是否可用 722
 - 24.5 处理64位Excel 723
 - 24.6 创建一个国际化的应用程序 724
 - 24.6.1 多语言应用程序 725
 - 24.6.2 VBA语言的考虑 726
 - 24.6.3 使用本地属性 726
 - 24.6.4 系统设置识别 727
 - 24.6.5 日期和时间设置 729
- 第25章 用VBA处理文件 731
 - 25.1 执行常见的文件操作 731
 - 25.1.1 使用与VBA文件相关的指令 732
 - 25.1.2 使用FileSystemObject对象 736

- 25.2 显示扩展文件信息 739
- 25.3 文本文件的处理 741
 - 25.3.1 打开文本文件 742
 - 25.3.2 读取文本文件 742
 - 25.3.3 编写文本文件 743
 - 25.3.4 获取文件序号 743
 - 25.3.5 确定或设置文件位置 743
 - 25.3.6 读写语句 744
- 25.4 文本文件操作示例 744
 - 25.4.1 导入文本文件中的数据 744
 - 25.4.2 将单元格区域的数据导出到文本文件中 746
 - 25.4.3 将文本文件的内容导入到单元格区域中 747
 - 25.4.4 记录Excel日志的用法 748
 - 25.4.5 筛选文本文件 749
 - 25.4.6 将单元格区域的数据导出为HTML格式 750
 - 25.4.7 将单元格区域导出到XML文件 752
- 25.5 压缩和解压缩文件 755
 - 25.5.1 压缩文件 755
 - 25.5.2 解压缩文件 757
- 25.6 使用ADO 758
- 第26章 使用Visual Basic组件 761
 - 26.1 IDE简介 761
 - 26.2 IDE对象模型 763
 - 26.3 显示VBA工程中的所有组件 766
 - 26.4 列出工作簿中的所有VBA过程 767
 - 26.5 用更新版本替换模块 768
 - 26.6 使用VBA编写VBA代码 770
 - 26.7 在设计时向用户窗体中添加控件 773
 - 26.7.1 设计时与运行时的用户窗体操作对比 773
 - 26.7.2 在设计时添加100个命令按钮 774
 - 26.8 通过编程创建用户窗体 776
 - 26.8.1 一个简单的运行时用户窗体示例 776
 - 26.8.2 一个有用(但是不简单)的动态用户窗体示例 778
- 第27章 理解类模块 783
 - 27.1 什么是类模块 783
 - 27.2 示例：创建NumLock类 784
 - 27.2.1 插入类模块 784
 - 27.2.2 给类模块添加VBA代码 785
 - 27.2.3 使用NumLockClass类 787
 - 27.3 更多有关类模块的信息 788
 - 27.3.1 对象属性编程 788
 - 27.3.2 对象的方法编程 790
 - 27.3.3 类模块事件 790
 - 27.4 示例：CSV文件类 791
 - 27.4.1 CSVFileClass的类模块级变量 791
 - 27.4.2 CSVFileClass的属性过程 791
 - 27.4.3 CSVFileClass的方法过程 792
 - 27.4.4 使用CSVFileClass对象 794
- 第28章 使用颜色 797
 - 28.1 指定颜色 797
 - 28.1.1 RGB颜色系统 798
 - 28.1.2 HSL颜色系统 798
 - 28.1.3 转换颜色 800
 - 28.2 理解灰度模式 802

28.3 颜色实验 803
28.4 理解文档主题 804
28.4.1 关于文档主题 804
28.4.2 理解文档主题颜色 805
28.4.3 显示所有的主题颜色 808
28.5 使用Shape对象 810
28.5.1 形状的背景色 810
28.5.2 形状和主题颜色 812
28.6 修改图表颜色 814
第29章 有关Excel编程的常见问题 819
29.1 获得关于FAQ的信息 819
29.2 Excel的一般问题 820
29.3 VBE 823
29.4 过程 825
29.5 函数 829
29.6 对象、属性、方法和事件 832
29.7 安全相关的问题 839
29.8 用户窗体 840
29.9 加载项 843
29.10 用户界面 845
第VII部分 附录
附录A VBA语句和函数引用 849
附录B VBA错误代码 857
附录C 本书的下载文件包中的内容 861
• • • • • ([收起](#))

[中文版Excel 2013高级VBA 编程宝典\(第7 版\)_ 下载链接1_](#)

标签

VBA

数据分析

excel

計算機

统计学

美国

技术流

会计

评论

好啰嗦

免费送书啦

[中文版Excel 2013高级VBA 编程宝典\(第7 版\) 下载链接1](#)

书评

[中文版Excel 2013高级VBA 编程宝典\(第7 版\) 下载链接1](#)