

中文版Excel2016高级VBA编程宝典



[中文版Excel2016高级VBA编程宝典 下载链接1](#)

著者:Michael Alexander

出版者:清华大学出版社

出版时间:2017-6

装帧:平装

isbn:9787302471066

借助VBA成为Excel高级用户
学习运用VBA语言的强大功能，将自己的Excel技能提升到全新水平！综合性指南

中文版Excel 2016高级VBA编程宝典(第8版)

将帮助你实现数据处理自动化，并开发出功能完备的Excel程序。本书针对Excel最新版本做了全面更新，内容新颖，助你开发大型和小型Excel应用。

本书直抒主题，以读者为中心，摒弃华而不实的、冗长的解释，呈现紧贴实用的示例、注意事项和诀窍，提供大量代码和示例模板，助你从普通数据分析人员蜕变为Excel编程专家。

作者介绍:

Michael Alexander是微软认证的应用开发人员，已撰写多本Microsoft Access和Excel书籍。Michael的个人网站是datapigtechnologies.com。

Dick

Kusleika在最近15年的时间里，一直通过在线论坛、博客、图书和会议等渠道传播知识，帮助用户深入了解和最大限度地运用Microsoft Office产品。

目录: 第 I 部分 Excel VBA基础知识

第1章 电子表格应用开发入门 3

1.1 关于电子表格应用 3

1.2 应用开发的步骤 4

1.3 确定用户的需求 4

1.4 对满足这些需求的应用
进行规划 5

1.5 确定最适用的用户界面 6

1.5.1 自定义功能区 7

1.5.2 自定义快捷菜单 7

1.5.3 创建快捷键 7

1.5.4 创建自定义对话框 8

1.5.5 在工作表中使用

ActiveX控件 8

1.5.6 开始开发工作 10

1.6 关注最终用户 10

1.6.1 测试应用 10

1.6.2 应用的安全问题 11

1.6.3 如何让应用程序看起来更
简明美观 13

1.6.4 创建用户帮助系统 13

1.6.5 将开发成果归档 14

1.6.6 给用户发布应用程序 14

1.6.7 在必要时对应用进行更新 14

1.7 其他开发问题 14

1.7.1 用户安装的Excel版本 15

1.7.2 语言问题 15

1.7.3 系统速度 15

1.7.4 显示模式 15

第2章 VBA概述 17

2.1 宏录制器 17

2.1.1 创建你的第一个宏 17

2.1.2 比较宏录制的绝对模式和
相对模式 20

2.1.3 关于宏录制的其他概念 24

2.2 Visual Basic编辑器概述 28

2.2.1 了解VBE组件	28
2.2.2 使用工程资源管理器	29
2.2.3 使用代码窗口	31
2.2.4 自定义VBA环境	33
2.2.5 “编辑器格式”选项卡	34
2.2.6 “通用”选项卡	35
2.2.7 “可连接的”选项卡	35
2.3 VBA的基础知识	36
2.3.1 了解对象	36
2.3.2 了解集合	37
2.3.3 了解属性	37
2.4 使用Range对象	40
2.4.1 找到Range对象的属性	41
2.4.2 Range属性	41
2.4.3 Cells属性	42
2.4.4 Offset属性	44
2.5 需要记住的基本概念	45
2.6 学习更多信息	46
2.6.1 阅读本书剩余的章节	47
2.6.2 让Excel来帮助编写宏	47
2.6.3 使用帮助系统	47
2.6.4 使用对象浏览器	47
2.6.5 从网上获取	48
2.6.6 利用用户论坛	49
2.6.7 访问专家博客	49
2.6.8 通过YouTube查找视频	50
2.6.9 通过Microsoft OfficeDev Center获取信息	50
2.6.10 解析其他的Excel文件	50
2.6.11 咨询周围的Excel人才	50
第3章 VBA编程基础	51
3.1 VBA语言元素概览	51
3.2 注释	53
3.3 变量、数据类型和常量	54
3.3.1 定义数据类型	55
3.3.2 声明变量	56
3.3.3 变量的作用域	58
3.3.4 使用常量	61
3.3.5 使用字符串	62
3.3.6 使用日期	62
3.4 赋值语句	63
3.5 数组	65
3.5.1 声明数组	66
3.5.2 声明多维数组	66
3.5.3 声明动态数组	66
3.6 对象变量	67
3.7 用户自定义的数据类型	68
3.8 内置函数	69
3.9 处理对象和集合	71
3.9.1 With-End With结构	72
3.9.2 For Each-Next结构	72
3.10 控制代码的执行	74
3.10.1 GoTo语句	74
3.10.2 If-Then结构	75

- 3.10.3 Select Case结构 78
- 3.10.4 指令块的循环 82
- 第4章 VBA的子过程 89
 - 4.1 关于过程 89
 - 4.1.1 子过程的声明 90
 - 4.1.2 过程的作用域 90
 - 4.2 执行子过程 91
 - 4.2.1 通过“运行子过程/用户窗体”命令执行过程 92
 - 4.2.2 从“宏”对话框执行过程 92
 - 4.2.3 用Ctrl+快捷键组合执行过程 93
 - 4.2.4 从功能区执行过程 94
 - 4.2.5 从自定义快捷菜单中执行过程 94
 - 4.2.6 从另一个过程中执行过程 95
 - 4.2.7 通过单击对象执行过程 98
 - 4.2.8 在事件发生时执行过程 99
 - 4.2.9 从“立即窗口”执行过程 100
 - 4.3 向过程中传递参数 100
 - 4.4 错误处理技术 103
 - 4.4.1 捕获错误 104
 - 4.4.2 错误处理示例 105
 - 4.5 使用子过程的实际示例 108
 - 4.5.1 目标 108
 - 4.5.2 工程需求 108
 - 4.5.3 已经了解的信息 108
 - 4.5.4 解决方法 109
 - 4.5.5 初步的录制工作 109
 - 4.5.6 初始设置 111
 - 4.5.7 代码的编写 112
 - 4.5.8 排序过程的编写 112
 - 4.5.9 更多测试 116
 - 4.5.10 修复问题 117
 - 4.5.11 实用程序的可用性 120
 - 4.5.12 对工程进行评估 121
- 第5章 创建函数过程 123
 - 5.1 子过程与函数过程的比较 123
 - 5.2 为什么创建自定义的函数 124
 - 5.3 自定义函数示例 124
 - 5.3.1 在工作表中使用函数 125
 - 5.3.2 在VBA过程中使用函数 125
 - 5.3.3 分析自定义函数 126
 - 5.4 函数过程 128
 - 5.4.1 函数的作用域 129
 - 5.4.2 执行函数过程 129
 - 5.5 函数过程的参数 131
 - 5.6 函数示例 132
 - 5.6.1 无参数的函数 132
 - 5.6.2 带有一个参数的函数 134
 - 5.6.3 带有两个参数的函数 136
 - 5.6.4 使用数组作为参数的函数 137
 - 5.6.5 带有可选参数的函数 138
 - 5.6.6 返回VBA数组的函数 140

- 5.6.7 返回错误值的函数 142
- 5.6.8 带有不定数量参数的函数 144
- 5.7 模拟Excel的SUM函数 145
- 5.8 扩展后的日期函数 148
- 5.9 函数的调试 149
- 5.10 使用“插入函数”对话框 150
 - 5.10.1 使用MacroOptions方法 151
 - 5.10.2 指定函数类别 152
 - 5.10.3 手动添加函数说明 153
- 5.11 使用加载项存储自定义函数 154
- 5.12 使用Windows API 154
 - 5.12.1 Windows API示例 155
 - 5.12.2 确定Windows目录 155
 - 5.12.3 检测Shift键 157
 - 5.12.4 了解有关API函数的更多信息 158
- 第6章 了解Excel事件 159
 - 6.1 Excel可以监视的事件类型 159
 - 6.1.1 了解事件发生的顺序 160
 - 6.1.2 存放事件处理程序的位置 160
 - 6.1.3 禁用事件 161
 - 6.1.4 输入事件处理代码 162
 - 6.1.5 使用参数的事件处理程序 163
 - 6.2 工作簿级别的事件 165
 - 6.2.1 Open事件 165
 - 6.2.2 Activate事件 166
 - 6.2.3 SheetActivate事件 166
 - 6.2.4 NewSheet事件 167
 - 6.2.5 BeforeSave事件 167
 - 6.2.6 Deactivate事件 167
 - 6.2.7 BeforePrint事件 168
 - 6.2.8 BeforeClose事件 169
 - 6.3 检查工作表事件 171
 - 6.3.1 Change事件 171
 - 6.3.2 监视特定单元格区域的修改 172
 - 6.3.3 SelectionChange事件 177
 - 6.3.4 BeforeDoubleClick事件 177
 - 6.3.5 BeforeRightClick事件 178
 - 6.4 监视应用程序事件 179
 - 6.4.1 启用应用程序级别的事件 180
 - 6.4.2 确定工作簿何时被打开 180
 - 6.4.3 监视应用程序级别的事件 181
 - 6.5 访问与对象无关联的事件 182
 - 6.5.1 OnTime事件 182
 - 6.5.2 OnKey事件 184
- 第7章 VBA编程示例与技巧 189
 - 7.1 通过示例学习 189
 - 7.2 处理单元格区域 190
 - 7.2.1 复制单元格区域 190
 - 7.2.2 移动单元格区域 191
 - 7.2.3 复制大小可变的单元格区域 191
 - 7.2.4 选中或者识别各种类型的

- 单元格区域 193
 - 7.2.5 调整单元格区域大小 194
 - 7.2.6 提示输入单元格中的值 195
 - 7.2.7 在下一个空单元格中输入一个值 196
 - 7.2.8 暂停宏的运行以便获得用户选中的单元格区域 197
 - 7.2.9 计算选中单元格的数目 199
 - 7.2.10 确定选中的单元格区域的类型 199
 - 7.2.11 有效地循环遍历选中的单元格区域 201
 - 7.2.12 删除所有空行 204
 - 7.2.13 任意次数地复制行 205
 - 7.2.14 确定单元格区域是否包含在另一个单元格区域内 206
 - 7.2.15 确定单元格的数据类型 207
 - 7.2.16 读写单元格区域 208
 - 7.2.17 在单元格区域中写入值的更好方法 209
 - 7.2.18 传递一维数组中的内容 211
 - 7.2.19 将单元格区域传递给Variant类型的数组 211
 - 7.2.20 按数值选择单元格 212
 - 7.2.21 复制非连续的单元格区域 213
- 7.3 处理工作簿和工作表 215
 - 7.3.1 保存所有工作簿 215
 - 7.3.2 保存和关闭所有工作簿 216
 - 7.3.3 隐藏除选区之外的区域 216
 - 7.3.4 创建超链接内容表 217
 - 7.3.5 同步工作表 218
- 7.4 VBA技巧 219
 - 7.4.1 切换布尔类型的属性值 219
 - 7.4.2 显示日期和时间 220
 - 7.4.3 显示友好时间 222
 - 7.4.4 获得字体列表 223
 - 7.4.5 对数组进行排序 224
 - 7.4.6 处理一系列文件 225
- 7.5 用于代码中的一些有用函数 227
 - 7.5.1 FileExists函数 227
 - 7.5.2 FileNameOnly函数 227
 - 7.5.3 PathExists函数 228
 - 7.5.4 RangeNameExists函数 228
 - 7.5.5 SheetExists函数 229
 - 7.5.6 WorkbooksOpen函数 229
 - 7.5.7 检索已经关闭的工作簿中的值 230
- 7.6 一些有用的工作表函数 231
 - 7.6.1 返回单元格的格式信息 232
 - 7.6.2 会说话的工作表 233
 - 7.6.3 显示保存或打印文件的时间 233
 - 7.6.4 理解对象的父对象 234

- 7.6.5 计算介于两个值之间的
单元格数目 235
- 7.6.6 确定行或列中最后一个
非空的单元格 236
- 7.6.7 字符串与模式匹配 237
- 7.6.8 从字符串中提取
第n个元素 238
- 7.6.9 拼写出数字 239
- 7.6.10 多功能函数 240
- 7.6.11 SHEETOFFSET函数 240
- 7.6.12 返回所有工作表中的
最大值 241
- 7.6.13 返回没有重复随机整数
元素的数组 242
- 7.6.14 随机化单元格区域 244
- 7.6.15 对单元格区域进行排序 245
- 7.7 Windows API调用 246
 - 7.7.1 理解API声明 246
 - 7.7.2 确定文件的关联性 247
 - 7.7.3 确定默认打印机的信息 248
 - 7.7.4 确定视频显示器的信息 249
 - 7.7.5 读写注册表 250
- 第 II 部分 高级VBA技术
- 第8章 使用透视表 255
 - 8.1 数据透视表示例 255
 - 8.1.1 创建数据透视表 256
 - 8.1.2 检查录制的数据
透视表代码 257
 - 8.1.3 整理录制的数据
透视表代码 258
 - 8.2 创建更复杂的数据透视表 260
 - 8.2.1 创建数据透视表的代码 261
 - 8.2.2 更复杂数据透视表的
工作原理 262
 - 8.3 创建多个数据透视表 263
 - 8.4 创建转换的数据透视表 266
- 第9章 使用图表 269
 - 9.1 关于图表 269
 - 9.1.1 图表的位置 269
 - 9.1.2 宏录制器和图表 270
 - 9.1.3 Chart对象模型 270
 - 9.2 创建嵌入式图表 271
 - 9.3 在图表工作表上创建图表 273
 - 9.4 修改图表 273
 - 9.5 使用VBA激活图表 274
 - 9.6 移动图表 275
 - 9.7 使用VBA使图表取消激活 276
 - 9.8 确定图表是否被激活 276
 - 9.9 从ChartObjects或Charts
集合中删除图表 277
 - 9.10 循环遍历所有图表 277
 - 9.11 调整ChartObjects对象的
大小并对齐 280
 - 9.12 创建大量图表 281

- 9.13 导出图表 283
- 9.14 修改图表中使用的数据 285
 - 9.14.1 基于活动单元格修改图表数据 286
 - 9.14.2 用VBA确定图表中使用的单元格区域 287
- 9.15 使用VBA在图表上显示任意数据标签 289
- 9.16 在用户窗体中显示图表 292
- 9.17 理解图表事件 295
 - 9.17.1 使用图表事件的一个示例 295
 - 9.17.2 为嵌入式图表启用事件 298
 - 9.17.3 示例：在嵌入式图表上使用图表事件 299
- 9.18 VBA制图技巧 301
 - 9.18.1 在整个页面上打印嵌入式图表 301
 - 9.18.2 创建未链接的图表 301
 - 9.18.3 用MouseOver事件显示文本 303
 - 9.18.4 滚动图表 305
- 9.19 使用迷你图 307
- 第10章 与其他应用程序的交互 311
 - 10.1 了解Microsoft Office自动化 311
 - 10.1.1 了解绑定概念 311
 - 10.1.2 一个简单的自动化示例 313
 - 10.2 从Excel中自动执行Access任务 314
 - 10.2.1 从Excel中运行Access查询 314
 - 10.2.2 从Excel运行Access宏 315
 - 10.3 从Excel自动执行Word任务 316
 - 10.3.1 将Excel数据传递给Word文档 316
 - 10.3.2 模拟Word文档的邮件合并功能 317
 - 10.4 从Excel自动执行PowerPoint任务 319
 - 10.4.1 将Excel数据发送到PowerPoint演示文稿中 319
 - 10.4.2 将所有Excel图表发送到PowerPoint演示文稿中 321
 - 10.4.3 将工作表转换成PowerPoint演示文稿 322
 - 10.5 从Excel自动执行Outlook任务 323
 - 10.5.1 以附件形式发送活动工作簿 324
 - 10.5.2 以附件形式发送指定单元格区域 324
 - 10.5.3 以附件形式发送指定的单个工作表 326

- 10.5.4 发送给联系人列表中的所有Email地址 327
- 10.6 从Excel启动其他应用程序 328
 - 10.6.1 使用VBA的Shell函数 328
 - 10.6.2 使用Windows的ShellExecute API函数 330
 - 10.6.3 使用AppActivate语句 331
 - 10.6.4 激活“控制面板”对话框 332
- 第11章 处理外部数据和文件 335
 - 11.1 处理外部数据连接 335
 - 11.1.1 手动创建连接 335
 - 11.1.2 手动编辑数据连接 338
 - 11.1.3 使用VBA创建动态连接 339
 - 11.1.4 遍历工作簿中的所有连接 340
 - 11.2 使用ADO和VBA来提取外部数据 342
 - 11.2.1 连接字符串 342
 - 11.2.2 声明记录集 343
 - 11.2.3 引用ADO对象库 344
 - 11.2.4 以编程方式使用ADO连接Access 344
 - 11.2.5 对活动工作簿使用ADO 345
 - 11.3 处理文本文件 348
 - 11.3.1 打开文本文件 348
 - 11.3.2 读取文本文件 349
 - 11.3.3 编写文本文件 349
 - 11.3.4 获取文件序号 349
 - 11.3.5 确定或设置文件位置 349
 - 11.3.6 读写语句 350
 - 11.4 文本文件操作示例 350
 - 11.4.1 导入文本文件的数据 350
 - 11.4.2 将单元格区域的数据导出到文本文件 351
 - 11.4.3 将文本文件的内容导出到单元格区域 352
 - 11.4.4 记录Excel日志的用法 353
 - 11.4.5 筛选文本文件 354
 - 11.5 执行常见的文件操作 354
 - 11.5.1 使用与VBA文件相关的指令 355
 - 11.5.2 使用FileSystemObject对象 359
 - 11.6 压缩和解压缩文件 361
 - 11.6.1 压缩文件 362
 - 11.6.2 解压缩文件 363
- 第III部分 操作用户窗体
- 第12章 使用自定义对话框 367
 - 12.1 创建用户窗体之前需要了解的内容 367
 - 12.2 使用输入框 367
 - 12.2.1 VBA的InputBox函数 368
 - 12.2.2 Excel的InputBox方法 370
 - 12.3 VBA的MsgBox函数 373
 - 12.4 Excel的GetOpenFilename

- 方法 377
- 12.5 Excel的GetSaveAsFilename
- 方法 380
- 12.6 提示输入目录名称 380
- 12.7 显示Excel的内置对话框 381
- 12.8 显示数据记录单 383
 - 12.8.1 使得数据记录单变得可以访问 383
 - 12.8.2 通过使用VBA来显示数据记录单 384
- 第13章 用户窗体概述 385
 - 13.1 Excel如何处理自定义对话框 385
 - 13.2 插入新的用户窗体 386
 - 13.3 向用户窗体中添加控件 386
 - 13.4 “工具箱”中的控件 387
 - 13.4.1 复选框 387
 - 13.4.2 组合框 388
 - 13.4.3 命令按钮 388
 - 13.4.4 框架 388
 - 13.4.5 图像 389
 - 13.4.6 标签 389
 - 13.4.7 列表框 389
 - 13.4.8 多页 389
 - 13.4.9 选项按钮 389
 - 13.4.10 RefEdit 389
 - 13.4.11 滚动条 390
 - 13.4.12 数值调节钮 390
 - 13.4.13 TabStrip 390
 - 13.4.14 文本框 390
 - 13.4.15 切换按钮 390
 - 13.5 调整用户窗体的控件 391
 - 13.6 调整控件的属性 393
 - 13.6.1 使用“属性”窗口 393
 - 13.6.2 共同属性 394
 - 13.6.3 满足键盘用户的需求 396
 - 13.7 显示用户窗体 397
 - 13.7.1 调整显示位置 398
 - 13.7.2 显示非模态的用户窗体 398
 - 13.7.3 显示基于变量的用户窗体 398
 - 13.7.4 加载用户窗体 399
 - 13.7.5 关于事件处理程序 399
 - 13.8 关闭用户窗体 399
 - 13.9 创建用户窗体的示例 400
 - 13.9.1 创建用户窗体 401
 - 13.9.2 编写代码显示对话框 403
 - 13.9.3 测试对话框 404
 - 13.9.4 添加事件处理程序 405
 - 13.9.5 完成对话框 406
 - 13.10 理解用户窗体的事件 407
 - 13.10.1 了解事件 407
 - 13.10.2 用户窗体的事件 408
 - 13.10.3 数值调节钮的事件 408
 - 13.10.4 数值调节钮与文本框

- 配套使用 410
- 13.11 引用用户窗体的控件 412
- 13.12 自定义“工具箱” 413
 - 13.12.1 在“工具箱”中添加新页 413
 - 13.12.2 自定义或组合控件 413
 - 13.12.3 添加新的ActiveX控件 415
- 13.13 创建用户窗体的模板 415
- 13.14 用户窗体问题检测列表 416
- 第14章 用户窗体示例 417
 - 14.1 创建用户窗体式菜单 417
 - 14.1.1 在用户窗体中使用命令按钮 417
 - 14.1.2 在用户窗体中使用列表框 418
 - 14.2 从用户窗体选中单元格区域 419
 - 14.3 创建欢迎界面 421
 - 14.4 禁用用户窗体的关闭按钮 423
 - 14.5 改变用户窗体的大小 424
 - 14.6 在用户窗体中缩放和滚动工作表 425
 - 14.7 列表框技巧 427
 - 14.7.1 向列表框控件中添加条目 428
 - 14.7.2 确定列表框中选中的条目 432
 - 14.7.3 确定列表框中的多个选中条目 433
 - 14.7.4 单个列表框中的多个列表 434
 - 14.7.5 列表框条目的转移 435
 - 14.7.6 在列表框中移动条目 436
 - 14.7.7 使用多列的列表框控件 438
 - 14.7.8 使用列表框选中工作表中的行 440
 - 14.7.9 使用列表框激活工作表 442
 - 14.7.10 通过文本框来筛选列表框 444
 - 14.8 在用户窗体中使用多页控件 446
 - 14.9 使用外部控件 448
 - 14.10 使标签动画化 450
- 第15章 高级用户窗体技术 453
 - 15.1 非模态对话框 453
 - 15.2 显示进度条 457
 - 15.2.1 创建独立的进度条 458
 - 15.2.2 集成到用户窗体中的进度条 461
 - 15.2.3 创建非图形化进度条 465
 - 15.3 创建向导 467
 - 15.3.1 为向导设置多页控件 468
 - 15.3.2 在向导用户窗体中添加按钮 468
 - 15.3.3 编写向导按钮的程序 468
 - 15.3.4 编写向导中的相关代码 470
 - 15.3.5 使用向导执行任务 471

- 15.4 模仿MsgBox函数 472
 - 15.4.1 模仿MsgBox函数：
MyMsgBox函数的代码 473
 - 15.4.2 MyMsgBox函数的
工作原理 474
 - 15.4.3 使用MyMsgBox函数 476
- 15.5 带有可移动控件的用户窗体 476
- 15.6 没有标题栏的用户窗体 477
- 15.7 使用用户窗体模拟工具栏 479
- 15.8 使用用户窗体来模仿
任务面板 481
- 15.9 可调整大小的用户窗体 482
- 15.10 用一个事件处理程序处理
多个用户窗体控件 485
- 15.11 在用户窗体中选择颜色 488
- 15.12 在用户窗体中显示图表 490
 - 15.12.1 将图表保存为GIF文件 491
 - 15.12.2 更改图像控件的Picture
属性 491
- 15.13 使用户窗体半透明 491
- 15.14 用户窗体上的数字推盘 493
- 15.15 用户窗体上的电动扑克 494

第IV部分 开发Excel应用程序

第16章 创建和使用加载项 497

- 16.1 什么是加载项 497
 - 16.1.1 加载项与标准工作簿的
比较 497
 - 16.1.2 创建加载项的原因 498
- 16.2 理解Excel的加载项管理器 500
- 16.3 创建加载项 501
- 16.4 加载项示例 502
 - 16.4.1 为加载项示例添加描述
信息 503
 - 16.4.2 创建加载项 503
 - 16.4.3 安装加载项 504
 - 16.4.4 测试加载项 505
 - 16.4.5 发布加载项 505
 - 16.4.6 修改加载项 505
- 16.5 比较XLAM和XLSM文件 506
 - 16.5.1 XLAM文件中的VBA
集合成员 506
 - 16.5.2 XLSM和XLAM文件的
可见性 507
 - 16.5.3 XLSM和XLAM文件的
工作表和图表工作表 507
 - 16.5.4 访问加载项中的VBA
过程 508
- 16.6 用VBA操作加载项 511
 - 16.6.1 向AddIns集合中添加项 511
 - 16.6.2 从AddIns集合中删除项 512
 - 16.6.3 AddIn对象属性 513
 - 16.6.4 作为工作簿访问加载项 516
 - 16.6.5 AddIn对象事件 516
- 16.7 优化加载项的性能 516

- 16.8 加载项的特殊问题 517
 - 16.8.1 确保加载项已经安装 517
 - 16.8.2 从加载项中引用其他文件 519
 - 16.8.3 为加载项检测适用的Excel版本 519
- 第17章 使用功能区 521
 - 17.1 功能区基础 521
 - 17.2 自定义功能区 522
 - 17.2.1 向功能区中添加按钮 523
 - 17.2.2 向快速访问工具栏中添加按钮 525
 - 17.2.3 自定义功能区的局限性 526
 - 17.3 创建自定义的功能区 527
 - 17.3.1 将按钮添加到现有的选项卡中 527
 - 17.3.2 向已有的选项卡中添加复选框 532
 - 17.3.3 功能区控件演示 535
 - 17.3.4 dynamicMenu控件示例 542
 - 17.3.5 关于自定义功能区的其他内容 544
 - 17.4 VBA和功能区 546
 - 17.4.1 访问功能区控件 546
 - 17.4.2 使用功能区 546
 - 17.4.3 激活选项卡 548
 - 17.5 创建老式工具栏 549
 - 17.5.1 Excel 2007及后续版本中老式工具栏的局限性 549
 - 17.5.2 创建工具栏的代码 549
- 第18章 使用快捷菜单 553
 - 18.1 命令栏简介 553
 - 18.1.1 命令栏的类型 553
 - 18.1.2 列出快捷菜单 554
 - 18.1.3 引用命令栏 555
 - 18.1.4 引用命令栏中的控件 555
 - 18.1.5 命令栏控件的属性 557
 - 18.1.6 显示所有的快捷菜单项 557
 - 18.2 使用VBA自定义快捷菜单 559
 - 18.2.1 快捷菜单和单文档界面 559
 - 18.2.2 重置快捷菜单 561
 - 18.2.3 禁用快捷菜单 562
 - 18.2.4 禁用快捷菜单项 562
 - 18.2.5 向“单元格”快捷菜单中添加一个新项 563
 - 18.2.6 向快捷菜单添加一个子菜单 565
 - 18.2.7 将快捷菜单限制到单个工作簿 567
 - 18.3 快捷菜单与事件 567
 - 18.3.1 自动添加和删除菜单 568
 - 18.3.2 禁用或隐藏快捷菜单项 568
 - 18.3.3 创建一个上下文相关的快捷菜单 568

第19章 为应用程序提供帮助	571
19.1 Excel应用程序的“帮助”	571
19.2 使用Excel组件的帮助系统	573
19.2.1 为帮助系统使用 单元格批注	573
19.2.2 为帮助系统使用文本框	574
19.2.3 使用工作表来显示 帮助文本	575
19.2.4 在用户窗体中显示 帮助信息	576
19.3 在Web浏览器中显示 “帮助”	580
19.3.1 使用HTML文件	580
19.3.2 使用一个MHTML文件	581
19.4 使用HTML帮助系统	582
19.4.1 使用Help方法来显示 HTML帮助信息	585
19.4.2 将“帮助”文件与应用 程序相关联	585
19.4.3 将一个帮助主题与一个 VBA函数相关联	586
第20章 理解类模块	589
20.1 什么是类模块	589
20.1.1 内置的类模块	590
20.1.2 自定义类模块	590
20.2 创建NumLock类	591
20.2.1 插入类模块	591
20.2.2 给类模块添加VBA代码	592
20.2.3 使用CNumLock类	593
20.3 属性、方法和事件编程	594
20.3.1 对象属性编程	594
20.3.2 对象的方法编程	595
20.3.3 类模块事件	596
20.4 QueryTable事件	596
20.5 创建存储类的类	599
20.5.1 创建CSalesRep和 CSalesReps类	599
20.5.2 创建CInvoice和 CInvoices类	601
20.5.3 用对象填充父类	602
20.5.4 计算佣金	603
第21章 兼容性问题	605
21.1 什么是兼容性	605
21.2 兼容性问题的类型	606
21.3 避免使用新功能	607
21.4 在Mac机器上是否可用	608
21.5 处理64位Excel	609
21.6 创建一个国际化应用程序	610
21.6.1 多语言应用程序	611
21.6.2 VBA语言的考虑	612
21.6.3 使用本地属性	612
21.6.4 系统设置识别	613
21.6.5 日期和时间设置	615
第 V 部分 附录	

[中文版Excel2016高级VBA编程宝典_ 下载链接1](#)

标签

- VBA
- Office
- 编程
- 计算机
- 会计
- excel
- EXCEL
- 美国

评论

有些舍近求远

[中文版Excel2016高级VBA编程宝典_ 下载链接1](#)

书评

[中文版Excel2016高级VBA编程宝典 下载链接1](#)