

职场特训

2007 Excel VBA 入门与提高实例大全

白领一族必学的Excel高级办公专业技能全书

国内外企业资深人士与一线专业培训师的经验之谈
紧扣职场工作的实际需求，让你学有所得，得有所用

钟 声 李远祥 郑开敏 编著

零基础学习，快速提高你的职场办公效率 ●

“职场特训”，让你在竞争中技高一筹 ●

代码注释解析，解决VBA理解与应用难题 ●

读者对象

有Excel基础，想要全面学习VBA开发的人员
想利用Excel VBA高效办公的人员
相关企事业单位的培训学员
需要一本案头必备查询手册的人员
Excel VBA编程爱好者

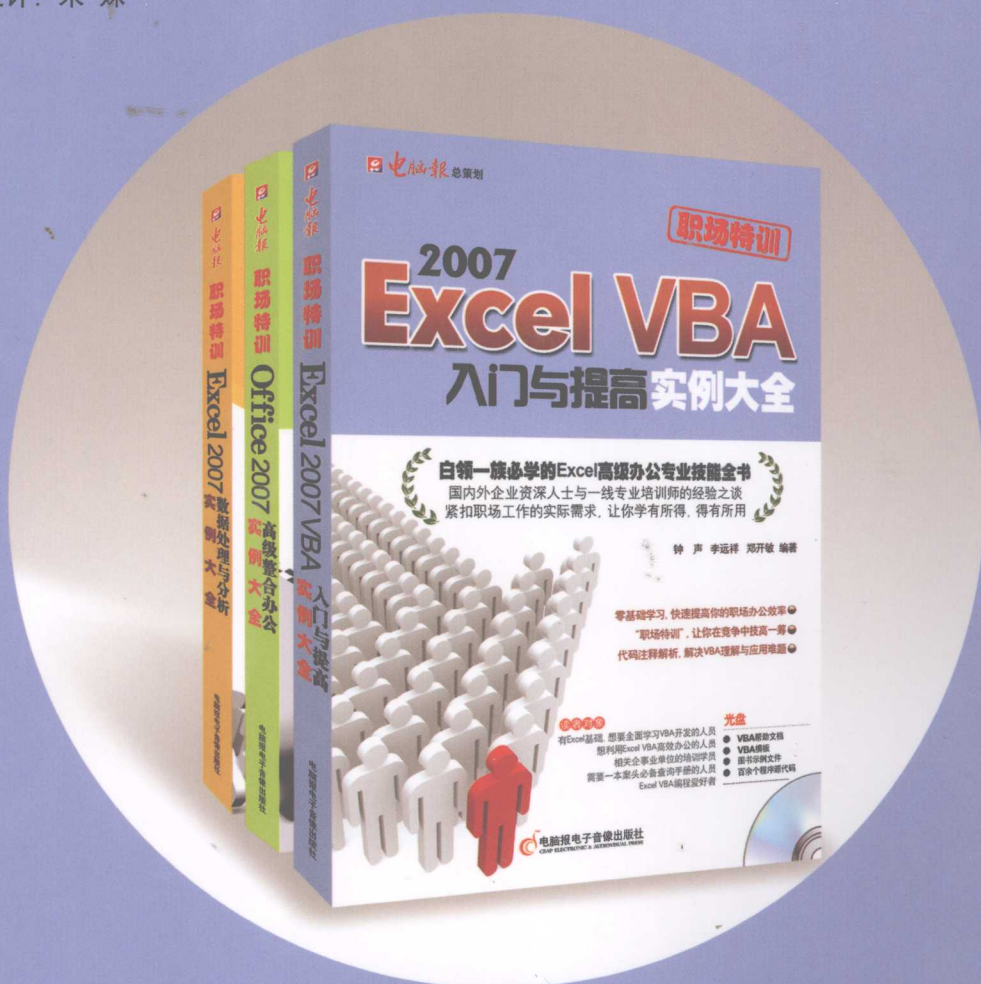
光盘

- VBA帮助文档
- VBA模板
- 图书示例文件
- 百余个程序源代码



电脑报电子音像出版社
CEAP ELECTRONIC & AUDIOVISUAL PRESS

封面设计：朱 姝



“职场特训” 系列

EXCEL 2007数据处理与分析实例大全
EXCEL 2007 VBA入门与提高实例大全
OFFICE 2007高级整合办公实例大全

本系列手册凝聚了国内多位Office专家的心血，这些专家都来自国内外大型企业的专业人士和一线的专业培训师。他们中的每一个人都具有多年Office应用经历，而且是其所在行业中的佼佼者。与普通的院校教程和自学手册不同，本系列手册紧扣企业的实际需求，以“职场特训”为特征，在教学中融入各种实际工作案例，由浅入深地介绍实际的工作技能，让你学有所得，得有所用。

分类建议：办公/Office

ISBN 978-7-89476-124-8



9 787894 761248 >

定价：45.00元 (1CD+手册)

支持
站点

IT精英网：www.itdaily.com.cn

电脑报书友会：www.itbook.com.cn

电脑报数位学院：www.cpcwedu.com

环球科学：www.sciam.com.cn

Excel 2007

VBA入门与提高

实例大全

钟 声
李远祥 编著

内容提要

本手册以快速帮助读者掌握 Excel 2007 VBA 的操作与应用为目的, 结合大量的实例, 深入浅出地逐步介绍 VBA 的基础知识和中高级应用, 使初学者能够快速掌握 VBA 的基本操作, 先从简单的创建宏开始, 再逐步介绍 VBA 编程基础、SUB 过程、function 过程、事件和对象以及与数据库之间的链接等知识, 再以综合案例的形式向读者展现 VBA 的魅力, 此外还为读者提供了上百个具有实用价值的 VBA 代码说明, 使读者可以很容易地理解这些程序并应用到实际工作中。

本书实例丰富, 可操作性强, 无论是初学者还是对 Excel 有一定了解的用户, 通过本书的学习都能轻松掌握 Excel 2007 VBA 功能的使用。本书适合企业办公人员自学, 也可作为培训教材使用。

光盘内容

- VBA帮助文档
- VBA模板
- 图书附录源代码
- 图片示例文件

版权所有 盗版必究

未经许可 不得以任何形式和手段复制和抄袭

Excel 2007 VBA入门与提高实例大全

编 著：钟 声 李远祥 郑开敏

技术编辑：黄 斌

版式设计：朱 姝

出版单位：电脑报电子音像出版社

地 址：重庆市双钢路3号科协大厦

邮政编码：400013

服务电话：(023)63658888

发 行：电脑报经营有限责任公司

经 销：各地新华书店、报刊亭

C D 生产：苏州新海博数码科技有限公司

文本印刷：重庆市联谊印务有限公司

开本规格：787mm×1092mm 1/16 27.5印张 500千字

版 号：ISBN 978-7-89476-124-8

版 次：2009年8月第1版 2009年8月第1次印刷

定 价：45.00元(1CD+手册)

前

言

在经济危机的洗礼下，各行各业都面临严峻的经济形势。很多企业单位为了降低成本，或多或少地进行了人力资源调整甚至裁员。一时间职场风雨飘摇，很多人感觉饭碗不保。在这个时候，过硬的工作能力和出色的工作技巧，能成为您在职场中站稳脚跟的有力武器。作为办公室白领，利用计算机进行各类数据、资料、表格处理乃至分析是一项必备的技能。但如何做得快、准、好，却又是考验一个人真实工作能力的试金石。

“职场特训”系列共有3个分册，分别对Office的职业应用和Excel的专业高级应用进行了教学和指导。与普通的院校教程和自学手册不同，本系列图书紧扣企业的实际需求，以“职场特训”为特征，融入各种实际工作案例，由浅入深地介绍实际的工作技能。

您选择了本书，说明您希望借助本书进一步提升自己的工作能力，获得更好的职场资源和空间，我们对你的信任表示感谢。这套书籍凝聚了国内多位Office专家的心血，这些专家都来自国内外大型企业的专业人士和一线的专业培训师。他们中的每一个人都具有多年Office应用经历，而且是其所在行业中的佼佼者。他们在日常工作中积累了大量的实战经验，发表了大量有关Office的绝妙应用方案。经过他们的共同努力，无数的用户在使用水平上获得了快速的提高。

“针对性强，易于上手，拿来就用”是本书最大特点。光盘中提供丰富的模板文件，不仅有手册中讲述的内容，还包含多个行业的经典应用模版，扩充了知识面，也延展了用户的学习范围。同时，本套产品在内容编排上尽量细致和人性化，以求读者能方便而又愉快地学习，力求实现专业“特训”的主题。

编者

2009年8月

目 录



第一章 Microsoft VBA 概述

1.1 什么是 VBA.....	2	1.2 第一个 VBA 程序	5
1.1.1 VBA 的历史	2	1.2.1 第一个 VBA 程序	5
1.1.2 VBA 与 Visual Basic 区别	2	1.2.2 执行 VBA 程序	7
1.1.3 VBA 的优点和功能	4	1.2.3 编辑 VBA 代码	8



第二章 VBA 编程基础

2.1 VBA 基础元素.....	12	2.3.1 Sub 过程	19
2.1.1 标识符	12	2.3.2 Function 函数	19
2.1.2 运算符	12	2.3.3 测试函数	20
2.1.3 数据类型	12	2.3.4 数学函数	20
2.1.4 变量与常量	13	2.3.5 字符串函数	21
2.1.5 数组	14	2.3.6 转换函数	21
2.2 常见语句	14	2.3.7 时间函数	22
2.2.1 注释和赋值语句	14	2.4 文件操作	22
2.2.2 书写规范	15	2.4.1 列表操作	22
2.2.3 判断语句	15	2.4.2 删除操作	23
2.2.4 循环语句	16	2.4.3 打开操作	23
2.3 过程和函数	19	2.4.4 读入操作	24
		2.4.5 写入操作	24



第三章 宏和控件的使用

3.1 宏的录制和保存	26	3.3.1 什么是控件	32
3.1.1 快速运行宏	26	3.3.2 常见控件	33
3.1.2 改变宏的保存位置	28	3.3.3 向工作表中添加控件	35
3.2 创建个人宏工作簿	29	3.3.4 设置控件的格式	37
3.2.1 保存个人宏	29	3.3.5 给控件命名	39
3.2.2 使用个人宏	30	3.3.6 使用VBA窗体	39
3.2.3 编辑个人宏	31	3.4 将宏指定给控件	42
3.3.4 删除个人宏	32	3.4.1 将宏指定给按钮	42
3.3 认识和使用 VBA 控件	32	3.4.2 将宏指定给图片	43
		3.4.3 将宏指定给工具栏按钮	45



第四章 常量和变量的使用

4.1 使用 Visual Basic 编辑器	48	4.2.3 明确变量的数据类型	59
4.1.1 使用 Visual Basic 编辑器的工具栏	48	4.2.4 给变量赋值	61
4.1.2 浏览代码	49	4.2.5 强制声明变量	64
4.1.3 使用开发手册	50	4.2.6 变量的范围	67
4.1.4 使用对象浏览器	52	4.2.7 静态变量	70
4.1.5 设置编辑器选项	53	4.2.8 声明和使用对象变量	71
4.2 变量的声明和使用	55	4.3 常量的使用	74
4.2.1 深入认识变量	55	4.3.1 在过程中使用常量	74
4.2.2 如何声明变量	55	4.3.2 内置常量	75



第五章 程序流程控制

5.1 判断语句	80	5.2.3 确定Case 子句里数值的范围	101
5.1.1 关系和逻辑运算符	80	5.2.4 在Case 子句里确定多个表达式	103
5.1.2 If...Then 语句	80	5.3 循环语句	104
5.1.3 多于条件的判断	85	5.3.1 DoWhile循环	104
5.1.4 If...Then...Else 语句	88	5.3.2 DoUntil循环	107
5.1.5 If...Then...ElseIf 语句	93	5.3.3 执行循环过程	110
5.1.6 嵌套的If...Then 语句	96	5.3.4 WhileWend 循环	111
5.2 选择语句	97	5.3.5 ForNext 循环	113
5.2.1 Select Case 语句	97	5.3.6 For EachNext 循环	115
5.2.2 和Case 子句一起使用Is	100	5.3.7 循环嵌套	117



第六章 子程序和函数

6.1 函数过程基础	120	6.2.6 使用可选的参数	131
6.1.1 关于函数过程	120	6.3 使用内置函数	133
6.1.2 创建函数过程	120	6.3.1 VBA内置函数介绍	133
6.2 执行函数过程	123	6.3.2 使用MsgBox 函数	134
6.2.1 从工作表里执行	123	6.3.3 MsgBox 函数的运行值	140
6.2.2 从另一个过程里执行	124	6.3.4 使用InputBox 函数	141
6.2.3 传递参数	126	6.3.5 数据类型转变	143
6.2.4 明确参数类型	128	6.3.6 使用InputBox的方法	145
6.2.5 按地址和按值传递参数	129	6.3.7 使用主过程和子过程	147



第七章 用 VBA 操纵外部对象

7.1 用函数控制外部对象	152	7.2.1 处理顺序文件	166
7.1.1 控制对象名称	152	7.2.2 利用 WSH 操作对象	175
7.1.2 控制对象属性	154	7.2.3 WSH 对象的方法和属性	178
7.1.3 操作文件夹对象	161	7.2.4 使用 WSH 运行应用程序	179
7.1.4 操作文件对象	162	7.2.5 使用 WSH 创建快捷方式	181
7.2 高级对象控制方法	166		



第八章 设计用户界面

8.1 使用用户窗体	184	8.2.4 数据有效性验证	197
8.1.1 为应用程序添加用户窗体	184	8.2.5 使用返回值	200
8.1.2 添加控件到用户窗体上	186	8.3 控制早期版本的工具栏和菜单	203
8.1.3 分配 Tab 键切换顺序	189	8.3.1 手工修改工具栏	203
8.1.4 为控件分配快捷键	190	8.3.2 创建自定义工具栏	205
8.2 实现窗体的自动功能	191	8.3.3 用 VBA 创建工具栏	206
8.2.1 初始化用户窗体中的值	191	8.3.4 手工修改菜单	208
8.2.2 显示用户窗体	194	8.3.5 通过 VBA 来使用菜单	210
8.2.3 控制窗体的行为	195		



第九章 程序的调试和排错

9.1 VBA 程序的调试	216	9.1.5 使用 STOP 进行中断	222
9.1.1 程序开发的测试和调试阶段	216	9.2 VBA 程序的排错	222
9.1.2 测试 VBA 过程	217	9.2.1 使用监视表达式	222
9.1.3 使用断点调试程序	218	9.2.2 使用快速监视	226
9.1.4 用即时窗口调试程序	221	9.2.3 使用本地窗口和堆栈	227



9.2.4 逐句运行VBA 过程 228

9.2.5 逐过程执行过程 229

9.2.6 使用条件编译 231

9.2.7 处理错误输出 233

9.2.8 创建集中的错误处理程序 239



第十章 Excel 对象事件编程

10.1 管理图表对象 244

10.1.1 创建图表 244

10.1.2 编辑录制的代码 245

10.1.3 执行修改后的图表代码 247

10.1.4 更复杂的图表示例 248

10.2 数据透视表对象 252

10.2.1 使用数据透视表 252

10.2.2 数据透视表应用 253

10.2.3 使用代码来创建数据透视表 254

10.2.4 查看录制的代码 256

10.3 Microsoft Office 对象 258

10.3.1 什么是自动化 258

10.3.2 引用自动化服务器 259

10.3.3 创建自动化服务器实例 259

10.3.4 使用VBA来控制 Word对象 260

10.4 Excel VBA 对象事件编程 264

10.4.1 事件过程介绍 265

10.4.2 激活和失活事件 266

10.4.3 事件次序 267

10.4.4 工作表事件 268

10.4.5 工作簿事件 274

10.4.6 图表事件 288

10.4.7 内嵌图表事件 291



第十一章 VBA 编程综合应用案例

11.1 办公用品管理系统 294

11.1.1 案例背景 294

11.1.2 创建入库及查询表格 294

11.1.3 建立按钮程序与公式 299

11.1.4 保护工作表 306

11.1.5 检查表单功能 309

11.1.6 建立出库及查询表格 312

11.1.7 创建管理窗口界面 322

11.1.8 添加界面程序 325

11.1.9 安全口令管理 327

11.1.10 设定宏运行环境 329

11.1.11 测试系统 330

11.2 俄罗斯方块程序 332

11.2.1 编程前准备 332

11.2.2 构建方块体 333

11.2.3 初始化程序 334

11.2.4 随机生成方块 336

11.2.5 方块进入和运动 337



11.2.6 判断方块所处环境	338	11.2.8 方块消除处理	341
11.2.6 移动操控的方块	339	11.2.9 杂项处理	342
11.2.7 方块的变形处理	339		



第十三章 Excel VBA 高级应用

12.1 Excel VBA 程序的类型	346	12.2.6 Excel工程的引用	354
12.1.1 Excel VBA 程序的类型	346	12.3 Excel 与 Access 的协作	355
12.1.2 部署VBA程序	346	12.3.1 使用Access对象库	355
12.1.3 管理VBA程序安全性	347	12.3.2 链接到Access 数据库	357
12.2 操控其他 Office 成员	348	12.3.3 使用ADO 链接数据库	361
12.2.1 启动其他Office组件	348	12.3.4 创建新Access 数据库	362
12.2.2 与其他Office组件交互	351	12.3.5 打开Access 窗体	364
12.2.3 反向操控Excel	352	12.3.6 运行Access 查询	366
12.2.4 Excel 自动化中的事件	353	12.3.7 从Access取数据到Excel	368
12.2.5 Excel 完成业务逻辑	354	12.3.8 Excel 数据导入Access	370



附录 Excel VBA 编程入门经典范例

一、Excel 应用程序对象及其常用方法	373	13. 改变 Excel 工作簿名字	377
1. 体验开 / 关屏幕更新	373	14. 使用 InputBox 方法	377
2. 使用状态栏	374	15. 设置页边距	377
3. 处理光标	374	16. 使用 Windows 的计算器	377
4. 获取系统信息	374	17. 在程序中运行另一个宏	378
5. 退出复制 / 剪切模式	375	18. 在指定的时间执行宏	378
6. 禁止弹出警告信息	375	19. 暂时停止宏运行	378
7. 将 Excel 全屏幕显示	375	20. 按下指定的按键后执行程序	378
8. Excel 启动的文件夹路径	375	21. 重新计算工作簿	379
9. 打开最近使用过的文档	376	22. 控制函数重新计算	379
10. 打开文件 (FindFile 方法)	376	23. 利用工作表函数	380
11. 文件对话框操作	376	24. 获取重叠区域	380
12. 保存 Excel 的工作环境	377	25. 获取路径分隔符	380
		26. 快速移至指定位置	380



27. 显示内置对话框	381
28. 退出 Excel	381
29. 关闭 Excel	381

二、窗口基本操作应用范例..... 381

1. 激活窗口	382
2. 窗口状态	382
3. 切换显示工作表元素	384
4. 显示公式	385
5. 显示 / 隐藏工作表标签	386
6. 命名活动窗口	386
7. 移动窗口到指定位置	386
8. 调整窗口	386
9. 拆分窗格	387
10. 冻结窗格	388
11. 设置网格线颜色	388
12. 设置工作表标签区域宽度和水平滚动条宽度比 例	388
13. 设置激活窗口时运行的程序	388
14. 获取指定窗口单元格区域地址	389
15. 返回指定窗口中所选择的工作表	389
16. 排列窗口	389
17. 窗口尺寸	390
18. 水平排列两个窗口	390
19. 改变窗口的高度和宽度	391
20. 移动窗口	392
21. 并排比较窗口	392
22. 返回或设置窗口中显示的视图	393
23. 返回窗口中可见单元格区域	393
24. 创建窗口	393
25. 设置窗口大小	393
26. 激活窗口	394
27. 滚动窗口	394
28. 测试所选单元格宽度和高度	395
29. 关闭窗口	396

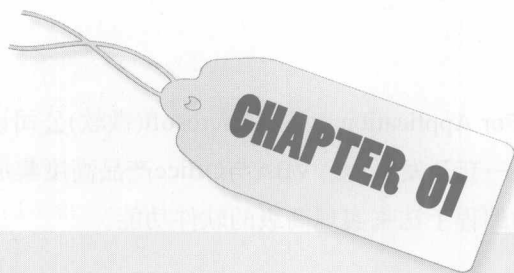
三、工作簿基本操作应用范例..... 396

1. 创建工作簿	396
2. 添加并保存新工作簿	397
3. 打开工作簿	397
4. 将文本文件导入工作簿中	397
5. 保存工作簿 (Save 方法)	398
6. 保存工作簿 (SaveAs 方法)	398

7. 取得当前打开的工作簿数 (Count 属性)	399
8. 激活工作簿 (Activate 方法)	400
9. 保护工作簿 (Protect 方法)	400
10. 解除工作簿保护 (UnProtect 方法)	401
11. 工作簿的一些通用属性示例	401
12. 访问工作簿的内置属性	401
13. 测试工作簿中是否包含指定工作表	403
14. 对未打开的工作簿进行重命名	404
15. 设置数字精度	404
16. 删除自定义数字格式	404
17. 控制工作簿中图形显示	404
18. 指定名称	405
19. 检查工作簿的自动恢复功能	405
20. 设置工作簿密码	405
21. 返回工作簿用户状态信息	406
22. 检查工作簿是否有密码保护	406
23. 决定列表边框是否可见	407
24. 关闭工作簿	407
25. 创建新的工作簿	408
26. 判断工作簿是否存在	409
27. 判断工作簿是否已打开	409
28. 备份工作簿	410
29. 从已关闭的工作簿中取值	412

四、工作表基本操作应用范例..... 414

1. 增加工作表	415
2. 复制工作表	415
3. 移动工作表	415
4. 隐藏和显示工作表	415
5. 获取工作表数	416
6. 获取或设置工作表名称	417
7. 激活 / 选择工作表	418
8. 获取当前工作表的索引号	418
9. 选取前一个工作表	419
10. 选取下一个工作表	419
11. 工作表行和列的操作	419
12. 工作表标签	421
13. 确定打印的页数	422
14. 保护 / 撤销保护工作表	422
15. 删除工作表	423
16. 判断一个工作表 (名) 是否存在	423
17. Application.WindowState 排序工作表	425



Microsoft VBA概述

在掌握了Microsoft Office中常用组件的基本使用方法之后,大部分用户会希望学习更高级的Office应用。这其中,Excel VBA是行业中应用最为普遍,但学习难度又较高的一项技术。很多用户希望学习Excel中关于VBA的各种知识,使得自己能轻松自如地驾驭Excel,从而在工作之中做到事半功倍。本章主要为VBA的初学者介绍VBA的概念、作用和特点。

1.1 什么是VBA

VBA的英文全称是Visual Basic For Application, 是由Microsoft(微软)公司设计开发, 并内置于Microsoft Office系列产品中的一套开发平台。VBA与Office产品高度集成, 用户利用简单的语法和规则, 通过简易快速的编程手法来实现高级的软件功能。

1.1.1 VBA的历史

在90年代早期, 使应用程序自动化还是充满挑战性的领域。对每个需要自动化的应用程序, 人们不得不学习一种不同的自动化语言。例如, 可以用Excel的宏语言使Excel自动化, 使用WordBasic使Microsoft Word自动化等等。而后, Microsoft决定让它开发出来的应用程序共享一种通用的自动化语言——Visual Basic for Application (VBA), 而不是使用不同的自动化语言, VBA便由此诞生了。

实际上, Visual Basic的应用程序版(VBA)是Microsoft公司长期追求的目标, 使可编程应用软件得到完美的实现。VBA作为一种新一代的标准宏语言, 具有上述跨越多种应用软件并且具有控制应用软件对象的能力, 使得程序设计人员仅需学习一种统一的标准宏语言, 就可以应用到特定的其它软件上去。程序设计人员在编程和调试代码时所看到的是相同的用户界面, 而且VBA与原应用软件的宏语言相兼容, 以保障用户在代码和工作上的投资。有了VBA以后, 多种应用程序共用一种宏语言, 节省了程序人员的学习时间, 提高了不同应用软件间的相互开发和调用能力。

在Office 2007中, 宏语言VBA适用于所有应用程序, 包括Word、Excel、PowerPoint、Access、Outlook以及Project等。早在Office 97的各种应用程序中, 就已经内置了Visual Basic编辑器, 这样, 用户无论是在Excel中, 还是在Word中以至是在Access中建立和管理VBA都具有统一的方法和标准。

1.1.2 VBA与Visual Basic区别

在中国, 绝大部分熟悉计算机的用户应该对Visual Basic(VB)并不陌生, VB是微软独创的一套面向对象的开发平台, 是由Basic发展而来的第四代语言。Visual Basic作为一套独立的Windows系统开发工具, 可用于开发Windows环境下的各类应用程序, 是一种可视化的、面向对象的、采用事件驱动方式的结构化高级程序设计语言。它具有高效率、简单易学及功能强大的特点。VB的程序语言简单、便捷, 利用其事件驱动的编程机制, 新颖易用的可视化设计工具, 并使用Windows应用程序接口(API)函数, 采用动态链接库

(DLL)、动态数据交换(DDE)、对象的链接与嵌入(OLE)以及开放式数据库访问(ODBC)等技术,可以高效、快速地编制出Windows环境下功能强大、图形界面丰富的应用软件系统。由于其简易和傻瓜化,深受入门级开发人员的喜爱,在90年代一度成为绝大部分院校计算机课程的标准内容之一。直到.net和java出现之后,它才慢慢淡出人们的视线,但是直到今天的.net开发平台中,依然保留了对VB语法的支持。

Microsoft
Visual Basic



Visual Basic程序很大一部分以可视(Visual)形式实现,这意味着在设计阶段就可以看到程序运行的屏幕画面,用户在设计时就能够方便地改动画面图像、大小、颜色等,直到满意为止。VB的用户可以是缺乏Windows及C语言开发经验的专业软件人员,也可以是具有Windows开发经验的专业人员,VB的可视化编程方法使得原来繁琐枯燥、令人生畏的Windows应用程序设计变得轻松自如、妙趣横生。以往的Windows应用程序开发工具在设计图形用户界面时,都是采用编程的方法,并伴随大量的计算任务,一个大型应用程序约有90%的程序代码用来处理用户界面,而且在程序设计过程中不能看到界面显示的效果,只有在程序运行时才能观察到,如果界面效果不佳,还需要回到程序中去修改。Visual Basic提供了新颖的可视化设计工具,巧妙地将Windows界面设计的复杂性封装起来,程序开发人员不必再为界面设计而编写大量程序代码,仅需采用现有工具按设计者要求的布局,在屏幕上画出所需界面,并为各图形对象设置属性即可。VB自动产生界面设计代码,这样便将事先编制好的控件可视地连接到一起,构成一个随时可调整的界面。

VBA不但继承了VB的开发机制,而且还具有与VB相似的语言结构,它们的集成开发环境IDE(Integrated Development Environment)也几乎相同。VB可运行直接来自Windows 9x或NT桌面上的应用程序,而VBA的项目(Project)仅由使用VBA的Excel、Word、PowerPoint等称为宿主(Host)的Office应用程序(Application)来调用。

那么总的来说VBA与VB有什么区别呢?

(1) Visual Basic是设计用于创建标准的应用程序,而VBA是用于使已有的应用程序自动化。

(2) Visual Basic具有自己的开发环境,而VBA必须“寄生于”已有的应用程序。

(3) 要运行Visual Basic开发的应用程序,用户不用在他的系统上访问Visual Basic,因为Visual Basic开发出的应用程序是可执行的。而由于VBA应用程序是寄生性的,执行它们要求用户访问“父”应用程序,例如Excel。

尽管存在这些不同,Visual Basic和VBA在结构上仍然非常相似。事实上,如果你已经了解了Visual Basic,会发现学习VBA非常快。相应地,学完VBA会给Visual Basic的学习打

下坚实的基础。而且,当学会在Excel中用VBA创建解决方案后,你就已经具备了在Word、Project、Access、Outlook、FoxPro和PowerPoint中用VBA创建解决方案的大部分知识。

1.1.3 VBA的优点和功能

在了解了VBA的基本概念之后,你也许希望知道可以用VBA干什么。在Excel中使用VBA可以实现的功能包括:

- (1) 使重复性的任务自动化。
- (2) 自定义Excel中工具栏、菜单和窗体的界面。
- (3) 简化模板的使用。
- (4) 为Excel环境添加额外的功能。
- (5) 创建报表。
- (6) 对数据执行复杂的操作和分析。

你以前也许没有想到将应用程序用作开发平台。大多数人考虑开发应用程序时,想到的都是像Visual Basic或者JAVA这样的程序语言。但是对于简单的数据处理业务来说,采用Excel作为开发平台有许多好处,这些好处包括:

- (1) Excel的应用程序功能强大,包括打印、文件处理、格式化和文本编辑。
- (2) Excel具有大量可供选择的内置函数。
- (3) Excel提供熟悉的界面。
- (4) 可连接到多种格式的数据库。

如果你以前曾经用某种语言编写过程序,你就会知道,你不得不费大量的精力去完成一些基本的功能,包括文件的打开和保存,以及剪贴板操作等等。而在Excel上进行开发则带来了使用应用程序开发解决方案的一个主要的优点——寄主应用程序已经具备了各种基本功能,需要做的只是使用它,如使用Excel中包括文件处理、文本编辑和格式化在内的各种功能。因为是在Excel中开发解决方案,所以还可以直接访问Excel的扩展函数库,如Excel用户所熟悉的所有函数(包括SUM、IRR、MAX、FV、PMT和AVG等等),这些函数在对于Excel开发者来说都是可用的。

从解决方案的最终用户的角度看,他们是在已经知道如何使用的应用程序上进行工作,因此可从中受益。他们对Excel的菜单系统、工具栏和工作表区域都很熟悉。正因为如此,他们会立刻对你用VBA开发的的自动化解决方案感到满意。

用Excel开发解决方案的其他一些优点也十分明显,例如Excel连接数据库的特征:如果在解决方案的窗体中需要对数据库(例如Microsoft SQL Server或者Microsoft Access)进行操作,由于Excel可以很容易地做到,因此,可以为你节省大量的精力。

1.2 第一个VBA程序

1.2.1 第一个VBA程序

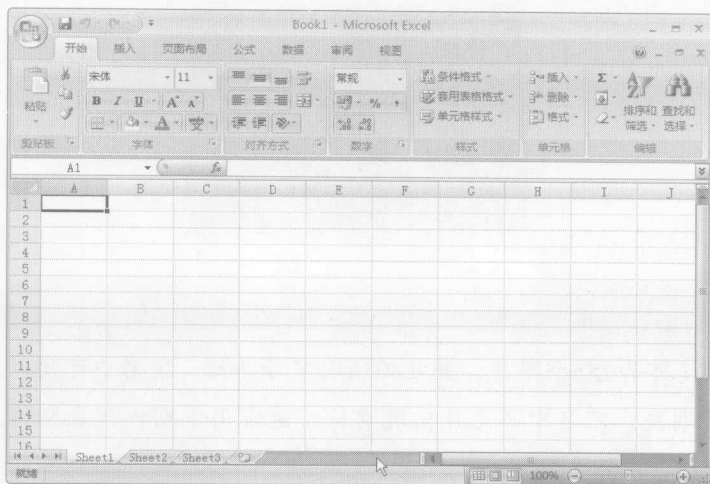
在介绍学习VBA代码编写之前,应该花几分钟录制一个宏。Excel的宏录制器允许记录一系列的操作,并且将这些操作转换为VBA代码。即使当你对编写VBA代码已经完全熟悉,也会在工作时使用到宏录制器。作为VBA开发者,使用宏录制器有两个原因:一个原因是因为使用宏录制器可以节省时间,开发者通常用它来建立应用程序的基础;另一个原因是宏录制器可以用作教学工具。如果你不能确定如何编写一系列的步骤,可以进行录制,再查看代码。

提示

对于常用的功能,Excel中都提供了相应的快捷键,这些快捷键均应牢牢记住,并尽可能经常使用。

以下将要录制的宏非常简单,只是改变单元格的字体和颜色。虽然有其他方法可以实现这种类型的任务(例如风格、自动套用格式等等),但是这一系列步骤为宏录制器提供了很好的示例。请完成如下步骤:

(1) 在Excel 2007环境中新建一个工作簿,确认所有其他工作簿已经关闭,以便能够很容易地对录制的宏进行定位和处理。



(2) 在单元格A1中输入你的姓名,在单元格B1中输入你的姓,在单元格C1中输入你居住的城市名,在单元格D1中输入你居住的国家名。以上操作将为下面的练习提供用来处