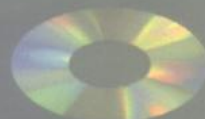




青松电脑系列丛书



附光盘

徐国平 郭力平 主编



VBA

中文版程序设计与应用



清华大学出版社

X 74

提高 Office 工作效率

VBA 中文版程序设计与应用

(Visual Basic for Applications 5)

徐国平 郭力平 主编

赵力平 李 斌 许 平 等编著

清华大学出版社

前 言

Visual Basic for Applications 5.0(简称 VBA 97)是新一代标准宏语言,是基于 Visual Basic for Windows 发展而来的。VBA 提供了面向对象的程序设计方法,提供了十分完整的程序设计语言。VBA 可以直接应用 Office 套装软件的各项强大功能,支持程序的开发和应用。VBA 易于学习掌握,允许用户通过宏记录器记录用户的各种操作并将其转换为 VBA 程序代码,这样,用户可以很容易地将日常工作转换为 VBA,使工作自动化。因此,对于在工作中需要经常使用 Office 套装软件的用户,学用 VBA 有助于提高工作效率。另外,对于程序设计人员,由于 VBA 可以直接应用 Office 套装软件的各项功能,所以应用程序的设计和开发更加方便快捷。

在 Office 各应用程序中提供一套通用的标准宏语言,乃是 Microsoft 长期追求的目标,目前终于在 Office 97 中得以实现。在 Office 97 中,宏语言 VBA 适用于所有应用程序,包括: Word、Excel、PowerPoint、Access、Outlook 以及 Project。在 Office 97 的各应用程序中,用户建立和管理 VBA 具有统一的方法和标准。

本书是作者基于长期应用研究与教学实践基础上写成的。本书也是目前国内为数不多的 VBA 97 专门书籍之一。

全书共分四篇计 20 章:第一篇 VBA 概论;第二篇 VBA 程序设计基础;第三篇 VBA 程序设计应用;包括 VBA 分别在 Excel、Word、PowerPoint、Access、Outlook 以及 Project 等中的应用,该编是本书的重点;第四篇 VBA 的高级应用:VBA 的 Web 应用、项目设计安全性、Help 文件、VBA 的数据库应用、Internet/Intranet 策略以及在企业项目开发中的应用。

本书的读者对象为 Office 的中、高级用户。

本书是一本实用教材,内容简明、图文并茂、实用性强。适于已有 Office 一定基础的广大读者、院校师生,特别是机关、企业从事办公自动化应用与开发的人员以及外企高级职员。本书可作为培训教材或工具用书。

本书由徐国平、郭力平主编,参加编写工作的还有:赵力平(第二篇)、李斌、许平、宋红欣等,徐一君、王小勇参加了助编、绘图以及校订工作。

在本书的编写过程中曾得到中国 UNIX 用户协会(CUUG)、美国 AT&T 公司系统开发工程师徐鹏力(PerryXu, AT&T)的支持和帮助,清华大学出版社与北京清华松岗电脑信息技术有限公司为本书迅速出版付出了辛勤劳动,谨此一并致谢!

限于编者水平,本书恐有疏漏不妥之处,尚祈读者不吝指正。

CUUG 应用研究室

1998 年 6 月

目 录

第一篇 VBA 概论

第 1 章 VBA 初步	(1)
1.1 Office 97 中的 VBA	(1)
1.1.1 什么是 VBA	(1)
1.1.2 Visual Basic (VB) 与 VBA	(1)
1.1.3 通用自动化语言	(2)
1.2 VBA 对象	(3)
1.3 对象的属性和方法	(4)
第 2 章 VBA 编辑器	(6)
2.1 Visual Basic 编辑器	(6)
2.2 工程资源管理器	(7)
2.3 属性窗口	(8)
2.4 代码窗口	(9)
2.5 记录宏	(11)
2.5.1 录制、查看、编辑和执行宏	(11)
2.5.2 录制宏实例	(13)
2.6 为宏安排菜单或按钮	(14)
2.7 用户窗体	(16)

第二篇 VBA 程序设计基础

第 3 章 VBA 编程基础	(21)
3.1 选取宿主应用程序	(21)
3.2 把代码放入项目的模块中	(21)
3.3 命名约定	(22)
3.4 VBA 过程	(22)
3.4.1 三种过程	(22)
3.4.2 建立过程	(22)
3.4.3 语句形式	(22)
3.4.4 语句元素	(23)

3.4.5 注释语句和赋值语句	(24)
第4章 数据类型、常量和变量	(25)
4.1 数据类型	(25)
4.2 常量	(26)
4.3 变量	(26)
4.3.1 声明变量	(26)
4.3.2 变体型变量	(27)
4.3.3 数组变量	(27)
4.3.4 多维数组变量	(28)
4.3.5 动态数组	(28)
4.3.6 变量的作用域	(28)
4.3.7 静态变量	(29)
4.4 内部函数	(29)
4.4.1 数学函数	(29)
4.4.2 字符串函数	(30)
4.4.3 日期和时间函数	(30)
4.4.4 逻辑测试函数	(31)
4.4.5 类型转换函数	(32)
4.5 自定义数据类型	(32)
4.6 内部对话框	(33)
4.6.1 MsgBox ()	(33)
4.6.2 InputBox ()	(36)
第5章 过程	(37)
5.1 子例程	(37)
5.2 函数	(37)
5.3 属性过程	(38)
5.4 过程的使用	(38)
5.5 过程的参数	(38)
5.5.1 参数传递方式	(38)
5.5.2 使用可选参数	(39)
5.5.3 使用不确定个数的参数	(40)
第6章 程序控制语句	(41)
6.1 If 语句	(41)
6.1.1 单行 If 语句	(41)
6.1.2 块结构的 If 语句	(41)

6.2	Select Case 语句	(43)
6.3	Do...Loop 语句	(45)
6.3.1	Do While...Loop 语句	(45)
6.3.2	Do Until...Loop 语句	(46)
6.3.3	Do...Loop While 语句	(47)
6.3.4	Do...Loop Until 语句	(48)
6.4	While...Wend 语句	(49)
6.5	For...Next 语句	(49)
6.6	For Each...Next 语句	(51)
6.7	With...End With 语句	(51)
6.8	Exit 语句	(52)
6.9	Go To 语句	(52)
第 7 章	各种控件	(54)
7.1	文本框控件(TextBox)	(54)
7.2	标签控件(Label)	(55)
7.3	复选框控件(CheckBox)	(55)
7.4	列表框控件(ListBox)	(56)
7.5	组合框控件(ComboBox)	(58)
7.6	框架控件(Frame)	(60)
7.7	选项按钮控件(OptionButton)	(60)
7.8	微调按钮控件(SpinButton)	(61)
7.9	命令按钮控件(CommandButton)	(62)
7.10	开关按钮控件(ToggleButton)	(63)
7.11	图片控件(Image)	(63)
7.12	在工具箱中增删工具	(64)
7.13	计时器控件(Timer)	(65)
第 8 章	调试程序和错误处理	(67)
8.1	错误的类型	(67)
8.1.1	编译错误	(67)
8.1.2	运行阶段错误	(67)
8.1.3	逻辑错误	(68)
8.2	三种模式	(68)
8.3	三个调试窗口	(68)
8.4	调试工具	(69)
8.5	中断模式	(70)
8.5.1	进入中断模式	(71)

8.5.2	监视表达式	(71)
8.5.3	快速监视	(72)
8.5.4	“立即”窗口	(72)
8.5.5	断点	(73)
8.5.6	使用 Stop 语句	(74)
8.5.7	在中断模式下的工作	(75)
8.5.8	调用堆栈	(76)
8.6	陷阱	(81)
8.6.1	陷阱的构成	(81)
8.6.2	On Error 语句	(82)
8.6.3	Err 对象	(82)
8.6.4	错误处理程序	(82)
8.6.5	Resume 语句	(82)
8.6.6	例题	(83)

第三篇 VBA 程序设计应用

第 9 章	VBA 在 Excel 中的应用	(84)
9.1	了解 Excel 对象	(84)
9.2	应用 Excel 的主要对象	(85)
9.2.1	应用当前应用程序对象	(85)
9.2.2	工作簿对象集和工作簿对象	(96)
9.2.3	应用工作表对象集	(104)
9.2.4	应用工作表对象	(105)
9.2.5	应用单元格区域对象	(106)
9.3	建立和应用用户窗体	(112)
9.4	创建自定义函数	(118)
第 10 章	VBA 在 Word 中的应用	(120)
10.1	了解 Word 对象	(120)
10.1.1	Word 对象	(120)
10.1.2	对象操作	(121)
10.2	Word 模板与 VBA 的结合	(122)
10.2.1	Word 模板的背景知识	(122)
10.2.2	向模板中加入 VBA 模块	(126)
10.2.3	解决矛盾	(129)
10.3	自动完成 Word 任务	(129)
10.3.1	在启动时显示打开对话框	(129)

10.3.2	新的检索标记工具	(130)
10.3.3	选择打印机工具	(131)
10.3.4	大纲转换工具	(136)
10.4	将 Word Basic 宏转化为 Visual Basic	(138)
10.4.1	如何进行转换	(138)
10.4.2	并非所有的宏都可以转换到 VBA 中	(139)
第 11 章	VBA 在 PowerPoint 中的应用	(141)
11.1	了解 PowerPoint 对象	(141)
11.1.1	PowerPoint 与 Excel 对象模型的异同	(141)
11.1.2	应用 PowerPoint 对象模型	(142)
11.1.3	逐步了解 PowerPoint VBA 方法	(146)
11.1.4	学习 PowerPoint 中的重要方法、对象和属性	(150)
11.2	定制 PowerPoint 任务	(154)
11.2.1	在幻灯片放映或编辑方式下运行 VBA	(154)
11.2.2	利用 VBA 或利用 PowerPoint 的内部代码	(154)
11.2.3	自动生成 PowerPoint 状态表	(155)
第 12 章	VBA 在 Access 中的应用	(160)
12.1	对象分类	(160)
12.2	数据库管理对象	(163)
12.2.1	DBEngine 对象	(163)
12.2.2	Workspace 对象	(164)
12.2.3	Database 对象	(167)
12.2.4	TableDef 对象	(169)
12.2.5	Relation 对象	(171)
12.2.6	QueryDef 对象	(173)
12.2.7	Recordset 对象	(174)
12.2.8	Property 对象	(181)
12.2.9	Container 对象	(182)
12.2.10	Document 对象	(183)
12.2.11	Fields 对象	(184)
12.2.12	Parameter 对象	(187)
12.2.13	Index 对象	(188)
12.2.14	User 对象	(189)
12.2.15	Group 对象	(190)
12.3	管理窗体的对象	(191)
12.3.1	Forms 对象	(191)

12.3.2	Form 对象	(192)
12.3.3	Control 对象	(193)
12.3.4	Module 对象	(194)
12.4	管理报表的对象	(194)
12.4.1	Reports 对象	(194)
12.4.2	Report 对象	(195)
12.5	系统控制对象	(196)
12.5.1	Application 对象	(196)
12.5.2	Screen 对象	(197)
12.5.3	Debug 对象	(198)
12.6	将 Access 宏转化为 Visual Basic 代码	(198)
第 13 章	VBA 在 Outlook 中的应用	(202)
13.1	Outlook 编程环境概述	(202)
13.2	比较 VBA 与 Visual Basic Scripting Edition	(202)
13.3	设计 Outlook 窗体	(203)
13.3.1	创建 Outlook 窗体	(203)
13.3.2	保存并发布窗体	(206)
13.4	对 Outlook 编程	(208)
13.4.1	使用 Script 事件	(209)
13.4.2	讨论事件过程	(212)
13.4.3	经常使用的 Outlook 对象	(213)
第 14 章	VBA 在 Project 中的应用	(217)
14.1	了解 Project 对象	(217)
14.1.1	Application 对象	(218)
14.1.2	Project 对象	(219)
14.1.3	List 对象	(220)
14.1.4	Task 对象	(221)
14.1.5	Resource 对象	(224)
14.1.6	Calendar 对象	(224)
14.2	在项目中使用 VBA	(224)
14.2.1	人工建立过程	(225)
14.2.2	创建自定义表	(227)
14.2.3	创建自定义筛选器	(228)
14.2.4	创建自定义视图	(230)
14.2.5	创建宏	(231)
14.2.6	将宏应用于多个项目	(234)

14.3 项目自动化	(235)
14.3.1 自动报表	(235)
14.3.2 使用自动执行过程	(239)

第四篇 VBA 的高级应用

第 15 章 VBA 在 Web 发布中的应用	(240)
15.1 VBA 宏与 Web 发布	(240)
15.2 Web 发布策略	(241)
15.2.1 动态更新	(241)
15.2.2 从文件中获取部分数据	(242)
15.2.3 格式化 Web 页	(244)
15.2.4 保留备份	(245)
15.2.5 文件位置	(245)
15.3 Web 页发布的模块设计	(245)
15.3.1 插入文件	(246)
15.3.2 格式化表格	(248)
15.3.3 保存文档	(249)
15.3.4 模块灵活性	(249)
15.3.5 Web 页更新	(252)
15.4 设计模块编辑 Web 页	(253)
15.4.1 搜索文件	(253)
15.4.2 编辑文件	(254)
15.5 发布 Web 页	(255)
第 16 章 VBA 项目的安全性	(256)
16.1 了解 VBA 安全措施	(256)
16.2 建立模块级密码保护	(257)
16.3 Access 数据库安全性	(259)
16.3.1 将数据库保存为 MDE 文件	(259)
16.3.2 隐藏数据库对象	(260)
16.3.3 为数据库设置密码	(260)
16.3.4 为数据库设置用户级安全性	(261)
16.4 调整权限	(270)
16.4.1 开放式访问	(270)
16.4.2 保护单个模块和其它对象	(271)

第 17 章 为 VBA 程序建立帮助文件	(272)
17.1 创建提示文本和“这是什么”帮助	(272)
17.1.1 创建屏幕提示	(272)
17.1.2 创建“这是什么”帮助	(272)
17.2 帮助文件的特点	(273)
17.2.1 跳转	(274)
17.2.2 搜索	(274)
17.2.3 弹出式定义	(276)
17.2.4 使用图片	(276)
17.3 建立帮助文件	(276)
17.3.1 构造帮助文件	(277)
17.3.2 美化帮助文件	(282)
17.3.3 测试和创建	(285)
17.3.4 其它帮助文件生成工具	(287)
17.4 将帮助文件加入应用程序	(288)
第 18 章 VBA 的数据库应用	(289)
18.1 何时使用外部数据	(289)
18.1.1 查询(Query)	(289)
18.1.2 数据访问对象	(290)
18.1.3 数据类型	(290)
18.2 MS Query —— 从 Word 和 Excel 中访问外部数据	(290)
18.2.1 使用 MS 查询(MS Query)	(290)
18.2.2 在 Word 中使用 MS Query	(291)
18.2.3 创建 MS Query	(293)
18.2.4 浏览记录的宏	(298)
18.2.5 在 Excel 中使用 MS Query	(298)
18.3 DAO 在企业开发中的应用	(300)
18.3.1 Microsoft Jet	(300)
18.3.2 ODBC	(301)
18.3.3 使用 DAO 开发解决方案	(301)
第 19 章 Internet 和 Intranet 策略	(311)
19.1 微软公司的 Internet 策略	(311)
19.1.1 Internet 服务器工具现状	(311)
19.1.2 Internet 服务器工具展望	(311)
19.2 微软公司的 Web 开发工具	(312)
19.2.1 Word 97 制作网页	(313)

19.2.2	利用 Access 97 制作网页	(324)
19.2.3	Microsoft FrontPage 98	(329)
19.3	网景公司的 Web 开发工具	(338)
19.3.1	Netscape Composer 概述	(338)
19.3.2	打开及关闭 Netscape Composer	(339)
19.3.3	Netscape Composer 网页制作	(340)
19.4	其它 Web 开发工具	(348)
19.5	Internet 解决方案	(348)
19.5.1	Web 站点设计	(348)
19.5.2	Web 数据库引擎	(349)
19.5.3	Web 站点的安全性	(349)
19.6	Intranet 解决方案	(350)
19.6.1	Intranet 规划	(350)
19.6.2	Intranet 内容	(352)
第 20 章	VBA 在企业中的开发与应用	(354)
20.1	企业解决方案	(354)
20.1.1	企业信息系统	(354)
20.1.2	使用 VBA 降低企业成本	(355)
20.1.3	专业系统开发	(356)
20.2	在线事务处理系统和决策支持系统	(356)
20.2.1	VBA 在在线事务处理系统中的应用	(357)
20.2.2	VBA 在决策支持系统中的应用	(358)
20.3	VBA 与其它编程语言	(359)
20.3.1	OLE 自动化	(360)
20.3.2	Visual Basic	(360)
20.3.3	C 和 C ++	(361)
20.3.4	ActiveX	(361)
20.4	企业系统中的分布式对象技术	(362)
20.4.1	分布式对象技术	(362)
20.4.2	Network OLE	(363)
20.4.3	多层系统	(364)
附录	本书所附光盘内容	(367)
参考资料		(368)

第一篇 VBA 概论

第 1 章 VBA 初步

1.1 Office 97 中的 VBA

1.1.1 什么是 VBA

Visual Basic for Applications (简称 VBA) 是新一代标准宏语言,是基于 Visual Basic for Windows 发展而来的。它与传统的宏语言不同,传统的宏语言不具有高级语言的特征,没有面向对象的程序设计概念和方法;而 VBA 提供了面向对象的程序设计方法,提供了相当完整的程序设计语言。VBA 易于学习掌握,用户可以使用宏记录器记录用户的各种操作并将其转换为 VBA 程序代码。这样用户可以容易地将日常工作转换为 VBA 程序代码,使工作自动化。因此,对于在工作中需要经常使用 Office 套装软件的用户,学用 VBA 有助于使工作自动化,提高工作效率。另外,对于程序设计人员,VBA 可以直接应用 Office 套装软件的各项强大功能,从而使程序设计和开发更加方便快捷。

1.1.2 Visual Basic(VB)与 VBA

VBA 是基于 Visual Basic 发展而来的,它们具有相似的语言结构。Visual Basic 是 Microsoft 的主要图形界面开发工具,VBA 5.0(亦即 VBA 97)则是 Visual Basic 5.0 的子集。

Visual Basic 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。Visual Basic(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。

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% 的程序代码是用来处理用户界面的,而且在程序设计过程中不能看到界面显示的效果,只有在程序执行时才能观察到,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的程序代码,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 VBA 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。但是,经过优化,VBA 专门用于 Office 的各应用程序。VB 可运行直接来自 Windows 95 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。

1.1.3 通用自动化语言

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,它作为一种通用的宏语言可被所有的 Microsoft 可编程应用软件所共享。

在没有 VBA 以前,一些应用软件如 Excel, Word, Access, Project 等都采用自己的宏语言供用户开发使用,但每种宏语言都是独立的,需要用户专门去学习,它们之间互不兼容,使得应用软件之间不能在程序上互联。因此,拥有一种可跨越多个应用软件,使各应用软件产品具有高效、灵活且一致性的开发工具是至关重要的。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,就可以转换到特定的应用软件上去。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。

经过 Microsoft 公司的长期努力,目前 Office 97 终于在各应用程序中提供了一套通用的标准宏语言。在 Office 97 中,宏语言 VBA 适用于所有应用程序,包括 Word, Excel, PowerPoint, Access, Outlook 以及 Project。在 Office 97 的各应用程序中,新增了 Visual Basic 编辑器。这样,用户无论是在 Excel 中,还是在 Word 中,甚至在 Access 中建立和管理 VBA 都具有统一一致的方法和标准。

1.2 VBA 对象

VBA 与传统宏语言的重要区别之一就在于它是面向对象的。对象是为了在 VBA 程序中管理数据和代码的方便提出来的。在 VBA 中,对象是封装有数据和代码的客体。例如,在 Excel 97 的工作表中,一个单一的单元格或单元格区域就是一个 Range 对象。用户可以在 VBA 程序代码中为 Range 对象赋值,或者使用封装在 Range 对象中的代码清除 Range 对象的数据。

在 Office 97 中有许多对象。例如,菜单栏、工具栏、Excel 工作簿、Word 文档、窗体、图形及图表等。在 VBA 中,各对象并不是孤立存在毫无关联的,而是彼此之间存在包含与被包含的关系。我们把包含有其它对象的对象称为容器(Container)。容器可以包含其它对象,也可以被更大的容器所包含。例如,表 1-1 和表 1-2 中分别列出了 Excel 与 Access 中常用的 VBA 对象。

表 1-1 Excel 中常用的 VBA 对象

对 象 名	说 明
Application	当前应用程序
Workbook	工作簿对象
Workbooks	当前打开的所有工作簿对象集
Worksheet	工作表对象
Worksheets	当前工作簿中所有工作表对象集
Range	由一个或多个单元格构成的单元格区域对象
MenuBar	菜单栏对象
ToolBar	工具栏对象

表 1-2 Access 中常用的 VBA 对象

对 象 名	说 明
Application	应用程序(即 Access 环境)
DBEngine	数据库管理系统
Debug	Debug 窗口对象
Forms	所有处于打开状态的窗体所构成的集合
Reports	所有处于打开状态的报表所构成的集合
Screen	屏幕对象
Docmd	使用此对象从 Visual Basic 中运行 Microsoft Access 操作

在 VBA 中,要访问一个对象,必需清楚该对象在对象体系中处于何种位置,然后通过对象访问运算符“.”,从包含该对象的最外层开始,由外及里逐次取其子对象,一直到达要访问的对象为止。例如在 Excel 中,要访问 ProductSales.XLS 工作簿中 Sheet1 工作表的 A1 单元格,应按以下格式书写:

```
Application.Workbooks("ProductSales.XLS").Worksheets("sheet1").Range("A1")
```

为了使代码更简洁,可以使用访问对象的简化格式,即如果某对象的父对象是当前对象,那么可以将其父对象以上的对象省略。例如,如果 ProductSales.XLS 工作簿是当前工作簿,那么要访问 ProductSales.XLS 工作簿中 Sheet1 工作表的 A1 单元格,应按以下格式书写:

```
Worksheets("sheet1").Range("A1")
```

在 VBA 中,把具有相同的属性和方法并在对象体系中同属一个层次的所有对象,划归为一个对象类。例如,在 Excel 中,工作表的每一个单元格区域都是一个 Range 对象。这些 Range 对象构成了一个 Range 对象类。

在 VBA 中,容器对象包含有许多子对象,我们把同属一个对象类的所有子对象称为对象集。例如,在 Excel 中,Workbooks 对象包含有许多工作表,所有这些工作表的集合构成了一个名为 Worksheets 的对象集。对象集也是对象,在 VBA 中所用到的许多对象都是对象集。

访问对象集中的某一成员,只需在对象集名称的后面跟上成员的名称。例如,要访问当前工作簿的 Sheet1 工作表成员,应按以下格式书写:

```
Worksheets("sheet1")
```

1.3 对象的属性和方法

对象的属性是封装在对象内部的数据,它反映了对象的特征。VBA 中的每一个对象都拥有许多属性。对象的大多数属性是可读的,但不是所有的属性都是可写的。

在 VBA 中,访问对象的属性需要用到“.”运算符,语法格式如下:

Object.Property

如果上面的格式出现在赋值语句的右边或者表达式之中,表示是读取某一对象的属性值;如果出现在赋值语句的左边,表示是改写某一对象的属性值。在 Excel 中,下面的两行代码分别是对 Sheet1 工作表中 A1 单元格内容的读和写:

```
MyValue = Worksheets("Sheet1").Range("A1").Value  
Worksheets("Sheet1").Range("A1").Value = MyValue
```

对象的方法是封装在对象中用来操作对象属性的代码段。这些代码段对于用户来说是不可见的,也是不可改写的。用户在使用对象的方法时,只需关心方法的名称及参数,无需关心方法的具体实现过程。

在 VBA 中,使用对象的方法需要用到“.”运算符,语法格式如下:

Object.Method Param1,Param2,...,Paramn

另外,许多方法都携带有参数(Param1,Param2,...,Paramn)。使用对象的方法有两种方式携带参数:一种方式是将参数用一对括号括起来,并且紧随方法名之后;另一种方式是将参数写在方法名之后,并用一个空格符把方法名和第一个参数隔开。例如,在 Excel 中要打开 Value.XLS 工作簿,应使用 Workbooks 对象的 Open 方法。该方法携带有 FileName 参数,格式如下:

```
Workbooks.Open FileName:="c:\GLP\Value.XLS"
```

例如:

在 Word 中,要打开 C:\My Documents\song.doc 文件,则打开的格式如下:

```
Documents.Open FileName:="C:\My Documents\song.doc"
```