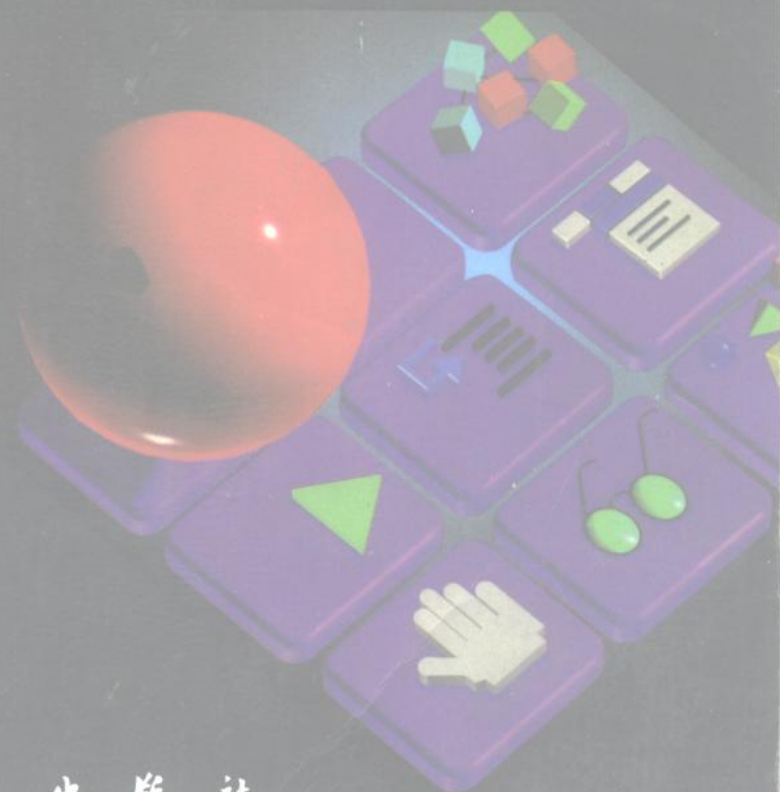


VISUAL BASICTM for Applications

程序设计大全



que[®]



科学出版社
龍門書局

Visual Basic for Applications 程序设计大全

用 Visual Basic 定制和开发 Excel

[美] Jeff Webb 著
沈伟 何军 于春燕 译
谢小兵 审校

科学出版社
龙门书局

1995

(京)新登字 092 号

内 容 简 介

Visual Basic for Applications 是 Microsoft 公司推出的新一代应用程序编程工具,它结合了 OLE Automation 技术,为 Windows 应用程序的编制提供了一种直观有效的方法。本书详细介绍了 Visual Basic for Applications 的特点、编程技术和实例,是 Visual Basic for Applications 编程的指导性书籍。

本书供从事软件应用与开发的技术人员阅读,也可供大中专院校有关专业的师生参考。

需要本书的读者,请直接与北京市海淀区 8721 信箱书刊部联系,邮编:100080,电话:2562329。

Jeff Webb

USING VISUAL BASIC FOR APPLICATIONS

Que Corporation, 1994

版 权 声 明

本书英文版名为《Using Visual Basic for Applications》由 Que Publishing 出版,版权归 Que Publishing 所有。本书中文版由 Simon & Schuster (Asia) Pte Ltd 授权出版。未经出版者书面许可,本书的任何部分均不得以任何形式或任何手段复制或传播。

Visual Basic for Applications 程序设计大全

[美] Jeff Webb 著

沈 伟 何 军 于春燕 译

谢小兵 审校

责任编辑 汪亚文

科学出版社 出版
龙 门 书 局

北京东黄城根北街 16 号

邮政编码:100717

双青印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1995 年 9 月第一版

开本:787×1092 1/16

1995 年 9 月第一次印刷

印张:40.875

印数:1—5000

字数:948 千字

ISBN 7-03-005076-2/TP·530

定价:48.80 元

谨以此书献给我的儿子。

作者简介

Jeff Webb 是微软公司 BASIC 文档小组的重要成员,他参加过 BASIC PDS, QBASIC, OLE Automation 和 Visual Basic 所有版本的工作。他曾为许多书撰过稿,并为许多帮助系统作出了贡献,尤其是: Visual Basic Reference; OLE 2 Programmer's Reference (Creating Programmable Applications) 和 Programming Integrated Solutions With Microsoft Office。现在,他经营 Seattle 的 Wombat 技术公司。

鸣 谢

在此我要感谢 Que 公司和 Microsoft 公司中那些帮助我完成这本书的人们。特别要指出的是, Lynn Northrap, 和 Russ Jacobs 为本书的提前完成作出了巨大贡献。Lynn 言词幽默,而业务熟悉,工作干得很漂亮。Russ 在项目结束前做了大量的额外工作。我再次向他们表示敬意。

还要感谢本书的技术编辑 Deirdre Maloy,是他使本书能适用于 Macintosh 机器的用户。

我还要感谢 Microsoft 公司的 Drew Fletcher, Glenn Hackney 和 Steven Mitchell。他们帮助我检查了 Excel 中有关 Visual Basic 的一些较模糊问题,并及时给予我一些最新的信息。我希望本书能使 Visual Basic 成为我们这个时代最优秀的编程语言。

最后,我要感谢 Microsoft 公司 Basic 编写组的主创人员对我的热情款待和帮助。

引 论

如今有很多可用的大型软件包,那么为什么那些非专业软件开发人员还要学习编程呢?将来的软件应用程序会不会变得不那么复杂?编程会不会成为过去呢?当使用应用程序或者操作系统命令的时候,你与计算机的交流要受到系统设计的控制。一个设计得非常好的系统看起来很自然,具有无限的能力,但是迟早你会发现它们的局限性,系统本身永远不能学会发展它自己。

当编程时,你学习与你的计算机进行合作。根据你的选择,系统可以是简单的,也可以是复杂的。这时,其局限性只由你的表达能力和你的硬件性能来决定。如果编写程序就像和老朋友交谈,而不是在填表格,那么这种人/机交互方式将会更加吸引人,令人愉快,最终使人上瘾。

Visual Basic 是可用于个人计算机上的功能最强大的交互式编程语言。我之所以这么说是因为使用 Visual Basic 和使用 C, C++, Pascal, FORTRAN, 或者 Lisp 相比,能用很少的时间和更多的乐趣做更多的事情。我以前用过那些语言,也知道它们也有一些值得推荐的长处——但在工作效率上、时间上、乐趣上,Visual Basic 表现得最为突出。

过去,现在和将来

当我开始为 Microsoft 工作时,Windows 正是一个令人烦恼的工具,在能运行 Excel 之前,必须要启动它;老练的用户知道其命令行,并且只有专业的程序员才知道汇编语言和 C。我被立刻指定负责一个名为 Basic PDS 的项目。在这里,我看到了一群疯狂的专注于一件事情的人们:击败 Borland。他们的 Turbo 产品销售量与我们 Microsoft 的产品相比逐渐地落在了后面,正是因为数量上的“竞争”才使得我们脱颖而出。

不久我们便推出了比以往的设计更加精练的快速的 Basic 开发环境,Borland 因此退出了 Basic 市场。与此同时,Windows 2.0 也被推出——我们注意到 Microsoft 产品的价格在上升,并开始开发另一个东西 Windows API。两位开发人员提出了一个非常大胆的设计:把正在为开发数据库产品而设计的宏语言作为 Basic 的 Windows 版本推出。管理部门称其为“机会”(一件好事)。正在市场人员争论其名字时,比尔·盖茨称其为“Visual Basic”。

Visual Basic 打破了所有的销售计划。最终,我们有了创建 Windows 应用程序的一种简单方法:当我在 Borland 公司的朋友菲利普·凯恩将 Visual Basic 比喻成“Crack cocaine”时,我才明白我们终于胜利啦。“写一个 VB 程序就行”已成为 Microsoft 的目标和准则。你需要刷房子吗?写一个 VB 程序就行。

虽然 Visual Basic 起源于一个数据库宏语言,但它成为了一个对应用程序几乎没有束缚的独立工具。要想达到比尔·盖茨的设计,Microsoft 需要一种重新组合 Visual Basic 应用程序的办法。Visual Basic 不能仅仅作为一个宏语言挤进每个应用程序;它还必须保留一个可以在应用程序间操作的开放编程工具来作为一个结合层。

当你在 Microsoft 说“集成”时,多数人都可能以为你在说“OLE!”,对象链接和嵌入

(OLE)是 Microsoft 处理一个应用程序包含另一个应用程序的文档的长远标准。最基本的, OLE 提供了一种应用程序间交换数据的方法。Visual Basic 小组承担了扩展 OLE 的任务, 这样你将能像交换数据那样交换命令。这些扩展被称为 OLE Automation 并在 OLE 2.0 中发布。

Excel 是第一个完全采用 OLE Automation 的应用程序,也是第一个含有 Visual Basic for Applications 的应用程序。Visual Basic for Application 完成了 Visual Basic 的使用——最终,有了一个编写应用程序的简单方法!

当然,这并不是结束。Visual Basic 将来要成为所有 Microsoft Office 应用程序的一部分。来自 Microsoft 或其它公司的其余软件也将采用 OLE 2.0 标准使其成为用 Visual Basic 可编的应用程序。不久,你将能使用 Visual Basic 在你的桌面型计算机上控制所有的应用程序。

不仅如此,OLE 2.0 还定义了一个远端过程控制标准,此标准使你能从一台桌面机上控制其它的计算机。这不仅仅是摧毁网络弱点的一个机会——它使你的 PC 能分配任务并控制大量的设备(包括打印机、电话、传真和复印机)。

我们知道——将来的某一天我们甚至能编程控制自己的 VCR。

关于编程风格

如果你不告诉我怎样写信,那我就不告诉你怎样编程(就这么简单)。关于“什么是清楚的,什么是不清楚的”我已有了—些看法,并在本书中我偶尔也讲了一些。优秀的编程风格是你所需要的也是你周围的人所需要的。

在使用变量之前声明所有的变量是一个好主意。另一个好主意是使用前缀来标识变量的数据类型和作用。在本书中的大多数示例代码都采用了这些原则;但是在代码中最清楚的片段在页面上往往并非最清楚。例如,在阅读程序主体之前,你要看大量的变量声明,这很令人生厌。因此,在一些地方我已做了让步。附录 D 详细讲述了有关编程风格的内容。

语法规约定

多数关于编程的书都使用了一些语法规约定。这种语法类型被称作 Backus-Naur Form (BNF)。下面的表格显示了当讨论函数、语句、方法和属性时所用的 BNF 标志法:

格式	描述
Abs(x)	以这种形式显示的词语表示该词是关键字,比如对象、属性、方法和函数,这种形式也用于在正文中表示过程和变量名。
<i>italic</i>	斜体表示该项必须提供,比如对象名和参数。方法参数中使用的名字与 Excel 表达这些参数的名字相同。
[]	方括号括起了可选项。
[item1 item2]	这表示两项之间选一个。

在这本书中,方法和属性的语法通过下面的格式来介绍:

Application. ActivateMicrosoftApp

Application. ActivateMicrosoftApp(index)

如果语法太长必须要折到另一行,那么在未结束行后要跟有特殊字符()表示该语法继续。

本书的组织

这本书被分成三部分:

- 第 1—4 章讲解了用 Visual Basic for Applications 编程的基础。
- 第 5—20 章描述了特殊的编程任务,并包含了对 Excel 对象、属性和方法的参考。
- 附录 A—E 包含了应用于多个编程问题的特殊信息的表格。

下面是关于每章内容和附录的简介:

第一章,“用 Excel 编程”告诉你如何使用 Visual Basic 编程工具在 Excel 中录制、修改和运行过程。这章还告诉你如何逐步建立一个过程,查找和改正错误并取得联机帮助。

第二章,“使用 Visual Basic”讲解 Visual Basic 中的基本编程结构并告诉你如何使用它们。这些基本结构包括过程、循环、变量、数组和错误处理。

第三章,“使用 Visual Basic 执行任务”告诉你如何使用 Visual Basic 语句、函数和操作符来处理各种数据类型。

第四章,“使用对象”说明如何使用 Excel 的对象、属性和方法来完成普通的编程任务。

第五章,“控制 Excel 应用程序”讲解如何使用 Application 对象来控制 Excel。这章还告诉你当 Excel 开始时如何自动运行过程,打开或关闭文件。

第六章,“控制 Excel 文件”告诉你如何使用 Workbook 和 AddIn 对象在 Excel 中打开和处理文件。

第七章,“控制 Excel 显示”告诉你如何使用 Window 和 Pane 对象来控制 Excel 窗口的位置和外观。

第八章,“使用工作表”告诉你如何使用 Sheets 集合和 Worksheet,Chart 及 Module 对象来操纵 Excel 工作簿中的工作表。

第九章,“使用单元格组”告诉你各种取得和使用 Range 对象的方法。这章还集中了使用单元格组的编程任务,比如查找和替换,以及取得和设置单元格值。

第十章,“链接和嵌入”告诉你如何使用 Picture 和 OLE Object 链接和嵌入来自其它工作簿或者其它应用程序的对象。

第十一章,“打印和显示结果”告诉你如何打印对象和如何使用 Style, Border, Font 和 PageSetup 对象来控制对象的屏幕输出和打印结果。

第十二章,“获取和操纵存储的数据”告诉你如何使用 PivotTable 对象和 Object Database Connectivity (ODBC) 来查看工作表中存储的数据和外部数据库。

第十三章,“创建图表”介绍了用于创建和操纵图表的对象,并告诉你如何使用多图表组。

第十四章,“控制图表”告诉你如何使用轴、图例和数据系列上的点。这章还告诉你如何自定义二维和三维图表的外观,以及如何向图表中添加垂直线、趋势线、网格线、误差线和其

它对象。

第十五章,“创建图形”解释了可以在图表和工作表上绘制的对象。图形对象包括 Arcs, Lines, Rectangle, Ovals 和 Pictures。

第十六章,“自定义菜单和工具栏”解释了 Menu 和 Toolbar 对象并告诉你如何自定义菜单和工具栏。这章还告诉你如何把自定义后的菜单和工作栏发送给其它用户。

第十七章“创建和显示对话框”解释了 Dialog 和 DialogSheet 对象并告诉你如何显示内置的 Excel 对话框,以及如何创建和显示你自己的自定义对话框。

第十八章,“使用其余的显示控件”解释了 Excel 提供的其余对话框控件对象,包括 Edit-Boxes, ListBoxes, OptionButtons 和 CheckBoxes。这章还涉及到了使用 ScrollBar 和 Spinner 对象。

第十九章,“发送和接收邮件”解释了 RoutingSlip 和 Mailer 对象并告诉你如何利用电子邮件来分发工作簿。这章还告诉你如何用电子邮件来分发和安装附件(add-ins)、模板和工具栏。

第二十章,“高级主题”告诉你如何创建和显示联机帮助,创建附件,使用 Visual Basic 3.0版下的 Excel 和使用动态链接库(DLLs)。

附录 A,“可捕获的错误”列出了可捕获的错误代码,并描述了捕获策略和处理方法。

附录 B,“键码表”列出了被 SendKeys 所用的特殊键码和被 Chr 和 Asc 函数所用的 ANSI 字符代码。

附录 C,“固有常量表”列出了分类的内置常量并给出它们的值。

附录 D,“专业水平代码的标准风格”列出了用于标识数据类型和变量范围的前缀,并描述了使你的程序更加可读的经验。

附录 E,“使用函数、属性和方法数据类型”列出了由 Visual Basic 函数和 Excel 属性和方法返回的值的的数据类型。这个附录还告诉你如何利用数据类型来节省内存和验证结果。

目 录

第一篇 Visual Basic for Applications 编程

第一章 用 Excel 编程	(2)
1.1 使用对象	(2)
1.2 录制代码	(4)
1.3 分析录制产生的代码	(6)
1.4 修改代码	(6)
1.5 改正错误	(8)
1.6 运行过程	(10)
1.7 单步调试代码	(11)
1.8 查看变量	(12)
1.9 在某处暂停代码	(13)
1.10 重新开始和停止代码的执行	(14)
1.11 用 Object Browser 查看对象和插入代码	(14)
1.12 取得帮助	(15)
1.13 Excel 的 Visual Basic 例子	(18)
第二章 使用 Visual Basic	(20)
2.1 第一个过程	(20)
2.2 在菜单中增加一个过程	(22)
2.3 控制程序流	(23)
2.4 把任务组织成过程	(30)
2.5 创建自己的方法和特性	(35)
2.6 创建变量和常量	(40)
2.7 创建用户定义的类型	(43)
2.8 创建并使用数组	(44)
2.9 处理错误(捕获和改正)	(45)
第三章 使用 Visual Basic 执行任务	(48)
3.1 创建表达式	(48)
3.2 使用 Visual Basic 运算符	(49)
3.3 处理数值	(56)
3.4 处理正文串	(59)
3.5 处理日期和时间	(66)
3.6 操纵数据文件	(67)
3.7 检查结果	(74)
3.8 与用户和应用程序的交互	(74)

第四章 使用对象	(78)
4.1 使用特性和方法	(79)
4.2 引用对象	(82)
4.3 为某项工作查找最恰当的对象	(86)
4.4 共有的属性和方法	(94)
4.5 全局属性和方法	(98)
4.6 使用 Excel 4.0 宏工作表函数	(100)
4.7 使用不能当作方法的宏工作表函数	(100)

第二篇 程序员参考手册

第五章 控制 Excel 应用程序	(110)
5.1 自动装载文件	(110)
5.2 自动运行过程	(111)
5.3 定制 Excel 应用程序	(114)
第六章 控制 Excel 文件	(147)
6.1 打开和保存文件	(147)
6.2 保护文件	(147)
第七章 控制 Excel 显示	(160)
7.1 打开和关闭一个窗口	(160)
7.2 翻滚一个窗口	(160)
7.3 窗口间切换	(161)
7.4 Pane 和 Panes 对象	(167)
7.5 AddIn 和 AddIns 对象	(168)
第八章 使用工作表	(170)
8.1 操纵工作簿中的工作表	(170)
8.2 处理单个工作表	(174)
8.3 操作 Visual Basic 模块	(190)
8.4 查看工作表	(193)
8.5 假设分析	(195)
第九章 处理单元格组	(198)
9.1 使用单元格组	(198)
9.2 访问多选范围内的区域	(240)
9.3 用名字引用范围	(241)
9.4 格式化及修改单元格内的正文	(244)
第十章 连接和嵌入	(247)
10.1 添加链接及图片	(247)
10.2 用 OLE 进行链接和嵌入	(252)
10.3 以正文框的形式嵌入正文	(260)

10.4	录音和放音.....	(267)
第十一章	打印和显示结果.....	(269)
11.1	打印和预览.....	(269)
11.2	运用自动化格式.....	(269)
11.3	控制样式.....	(270)
11.4	控制边框.....	(274)
11.5	控制字体.....	(277)
11.6	控制页面设置.....	(279)
第十二章	获取和操纵存储的数据.....	(286)
12.1	用数据透视表查看数据.....	(286)
12.2	以程序形式使用数据透视表.....	(292)
12.3	以程序的形式使用数据透视字段.....	(298)
12.4	以程序的形式使用数据透视项.....	(303)
12.5	用 XLODBC.XLA 检索数据	(304)
第十三章	创建图表.....	(317)
13.1	使用 Chart Wizard	(317)
13.2	在工作表中嵌入图表.....	(318)
13.3	生成并操纵图表.....	(323)
13.4	生成图表标题.....	(346)
13.5	使用多图表分组.....	(347)
第十四章	控制图表.....	(352)
14.1	在图表上绘制数据序列.....	(352)
14.2	控制轴.....	(360)
14.3	生成轴标题.....	(365)
14.4	控制数据标志.....	(367)
14.5	控制轴上的位置标志.....	(370)
14.6	从序列中取单个点.....	(370)
14.7	控制图表区域.....	(373)
14.8	控制绘图区域.....	(374)
14.9	控制图表的图例.....	(375)
14.10	改变图例中的文字	(377)
14.11	改变图例符	(377)
14.12	生成趋势线	(379)
14.13	向堆积柱形图或竖条图中添加序列线	(382)
14.14	向线形图表中添加高低线	(383)
14.15	给轴增加网格线	(384)
14.16	向线形图和面积图中增添垂直线	(386)
14.17	向序列中添加误差线	(387)
14.18	创建开/高/低/关线形图表.....	(388)

14.19	控制三维图表的显示	(389)
第十五章	创建图形	(391)
15.1	使用 Excel 的绘图工具	(391)
15.2	使用图形对象	(392)
15.3	将对象分组为一个单元	(397)
15.4	改变内部颜色和图案	(398)
15.5	画弧	(400)
15.6	绘制手画图形	(404)
15.7	绘制线条和箭头	(409)
15.8	画圆和椭圆	(412)
15.9	插入图片	(416)
15.10	绘制矩形	(416)
第十六章	自定义菜单和工具栏	(421)
16.1	编辑菜单	(421)
16.2	将过程加到菜单项中	(422)
16.3	保存和分配菜单	(422)
16.4	动态生成和显示菜单栏	(423)
16.5	动态生成和显示菜单	(425)
16.6	在菜单中增加菜单项	(427)
16.7	生成和编辑工具栏	(430)
16.8	动态生成和显示工具栏	(432)
16.9	动态地在工具栏中增加按钮	(435)
第十七章	创建和显示对话框	(447)
17.1	使用内置对话框	(447)
17.2	生成用户对话框	(461)
17.3	在代码中使用对话框	(469)
17.4	使用 DialogFrame	(484)
17.5	使用按钮运行过程	(486)
17.6	用标签显示不可编辑的数据	(492)
第十八章	使用其余的显示控件	(497)
18.1	用编辑框获取和显示数据	(497)
18.2	用复选框获得 True/False 值	(502)
18.3	用单选钮获得 True/False 值	(506)
18.4	用分组框将控件分组	(511)
18.5	用下拉式列表框显示列表	(515)
18.6	用列表框显示列表	(522)
18.7	使用滚动条	(529)
18.8	使用微调控件项	(533)
第十九章	发送和接收邮件	(538)

19.1	从 Excel 中发送邮件	(538)
19.2	分配其他类型文件	(540)
19.3	从代码中发送附加的文件	(543)
19.4	使用 Routing Slips	(543)
19.5	使用 Macintosh PowerTalk Mailers	(546)
第二十章	高级主题	(548)
20.1	创建联机帮助	(548)
20.2	显示联机帮助	(551)
20.3	使用 Add_ins	(553)
20.4	创建 Add_ins	(563)
20.5	分配 Add_in 文件	(563)
20.6	使用 AddIn 对象	(567)
20.7	从 Visual Basic 3.0 使用 Excel	(568)
20.8	使用动态链接库	(572)
20.9	总结	(576)

第三篇 附 录

附录 A	可捕获的错误	(578)
A.1	汇集错误	(578)
A.2	使用错误处理器	(579)
A.3	将错误处理作为一个捷径	(581)
A.4	程序员自定义的错误代码	(581)
A.5	可捕获的错误代码	(582)
附录 B	键码表	(586)
附录 C	固有常量表	(598)
附录 D	专业水平代码的标准风格	(624)
D.1	标识作用域	(624)
D.2	标识数据类型	(626)
D.3	选择有意义的名字	(631)
D.4	格式代码	(631)
D.5	注释代码	(632)
D.6	变量声明	(633)
D.7	总结	(633)
附录 E	使用函数、属性和方法数据类型。	(635)
E.1	基本数据类型的优点	(636)
E.2	将基本数据类型与变体型一起使用	(637)
E.3	函数,属性和方法数据类型	(637)

第一篇

Visual Basic for Applications 编程

1. 用 Excel 编程
2. 使用 Visual Basic
3. 使用 Visual Basic 执行任务
4. 使用对象

第一章 用 Excel 编程

多年来,应用程序已经包含了能使你自动完成各种日常任务的宏语言。虽然许多宏语言其功能很强,但它们都有几个致命的缺点:

- 每个语言都不相同。即使在一个公司卖出的应用程序中,也没有两个相似的宏语言。这样使其很难在应用程序间移植和再用。
- 宏语言往往被用户界面所束缚——当用户界面发生了变化,语言也发生了变化。你不得不重新学习每个新的版本。
- 随着应用程序的成长,语言变得更加复杂和难于理解。
- 没有一个简单的办法能从一个应用程序使用另一个应用程序,并绝对没有办法运行来自另一个应用程序中的宏。

Visual Basic for Applications 则逐渐解决了这些问题:

- Visual Basic 已被全世界超过 300000 用户公认为 Windows 编程标准。Visual Basic for Applications 坚持 Visual Basic 标准,并且 Microsoft 公司准备让其他的 Office 程序也提供 Visual Basic for Application。Excel 是第一个带有 Visual Basic 的应用程序。
- Visual Basic for Applications 使用了一个标准的“核心语言”和由应用程序提供的对象。这些对象体现了应用程序的基本特征,因此它们不可能改变。
- 由对象特征构成的层次结构要比由成千上万条宏语句组成的“平面”宏语言更加易于理解。
- Visual Basic for Applications 可以调用其他 Visual Basic for Applications 程序,即使它们在不同的应用程序中编写。同时,与多个应用程序工作的支持同时在 Visual Basic for Applications 的底层加以实现。

Visual Basic for Applications 在 Excel 中并不只是一个宏语言,它可以用来创建应用程序,解决商业问题和自动完成日常任务,就象搭积木一样,它是一个编程语言。

在这章中,你将学会如何完成下列工作:

- 使用对象控制 Excel。
- 录制和修改 Visual Basic 过程。
- 运行过程。
- 通过单步调试和观察变量弄清楚一个过程的内部工作。
- 在 Visual Basic for Applications 中获得联机帮助。

1.1 使用对象

Visual Basic 在 Excel 中用对象来获取信息和执行动作。对象是 Excel 应用程序的积木块;他们包含 Excel 的可见方面,比如工作表,以及和看不见的概念,比如页面设置(见图 1-1)。关于 Excel 对象的完整描述和怎样使用它们,见第四章。

一些对象的例子包括:

对象	描述
Application	Excel 的运行备份
WorkBook	一个完整的工作簿——一个 Excel 文件
Chart	在工作簿中的一个图形图表
WorkSheet	在工作簿中的一个完整的工作表
Range	在一张工作表上的一个单元格或格范围

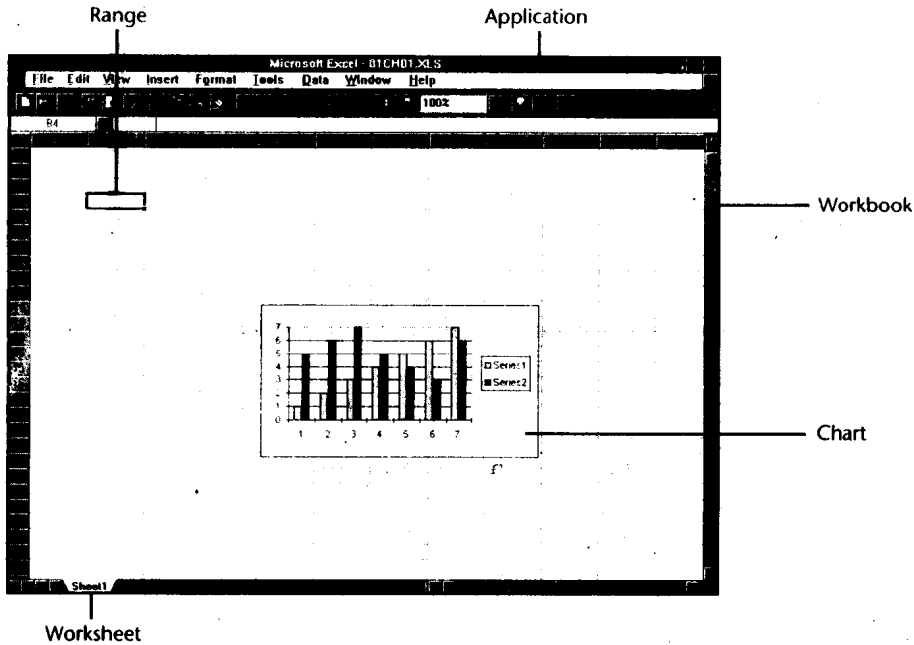


图 1-1 一些 Excel 对象

Excel 总共大概有 80 个对象。每个对象都允许你控制 Excel 的某个特性。比如，假设有了一个对象，它代表工作表上的单元格 A1(见图 1-2)。

在 Excel 中单元格 A1 是一个 Range 对象。工作表上的每个单元格都是一个 Range 对象，并且这些对象也可能包含多个单元格。比如，Range("A1:B4")包含了在 A1:B4 范围内的所有单元格。

使用 Visual Basic for Application 可以对对象做如下事情：

- 设置对象属性的值(见后面的属性定义)。下面的代码行在单元格中插入了一个公式：

```
Range("A1").Formula="Sum(B1,C1,C3)"
```

- 返回对象属性的值。下面的代码行取得了单元格的数字值：

```
X=Range("A1").Value
```

- 调用对象的方法。下面的代码行选定了单元格 A1：

```
.Range("A1").Select
```

通常，属性控制对象的外观，而方法则执行关于对象的动作。实际上，这种区分并不很清晰。属性和方法的真正的不同是语义：属性可以出现在等号(=)的左边或右边；方法只能出

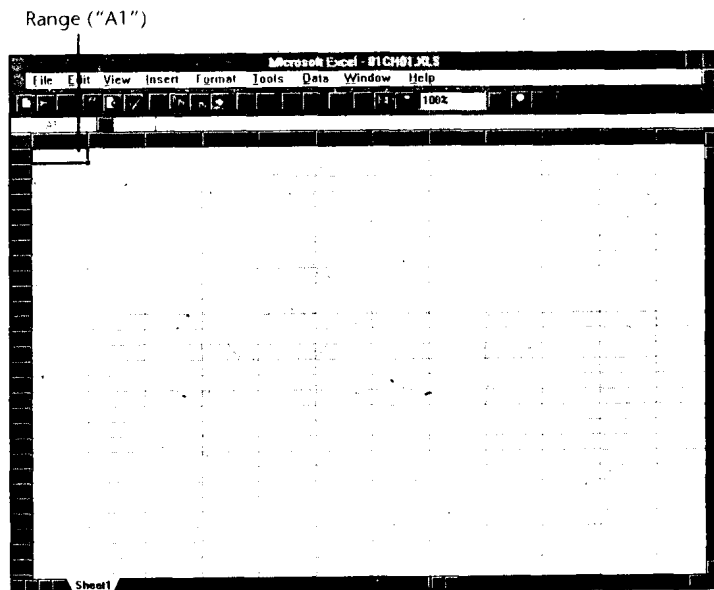


图 1-2 单元格 A1

现在等号的右边。

下面的代码行使用 Value 属性来设置单元格 A1 的值：

```
Range("A1").Value = 97
```

下行用 Address 方法返回单元格 A1 的地址：

```
x = Range("A1").Address()
```

注意：由于属性和方法之间的不同与每个属性或方法执行的任务相比是微不足道的，所以本书在描述对象时把属性和方法结合在一起。

每个对象都有许多不同的属性和方法——Range 对象就有 88 个方法和 46 个属性！这些属性和方法中的一部分是和其他对象共有的，但大部分是 Range 对象所独有的。Excel 大约有 1400 个可通过 Visual Basic for Application 使用的属性和方法。幸运的是，为了使学习编写程序更加容易，Excel 做了如下事情：

- 首先也是最重要的，你可以使用 Visual Basic 录制你在 Excel 中的动作，然后再查看和重演 Excel 根据这些动作产生的代码。
- 其次，Excel 允许你一次单步调试一行代码。由于可以看到每行代码的执行效果，因而可以迅速地把属性和方法同动作联系起来。
- 最后，Excel 为多数的 Visual Basic 语言元素提供上下文相关的联机帮助。有些元素的帮助是暗的或未启动的，但是通常 Help 都是可用的。关于 Visual Basic 核心语言的帮助（基本语句像 Do...Loop, If...Then, Dim 等等）尤其有用。Visual Basic 的核心语言是所有应用程序共有的并包含在一个单独的库中（对于 U. S. 版本是 VBAEN.DLL），该库被支持 Visual Basic 的所有其他程序所共享。