

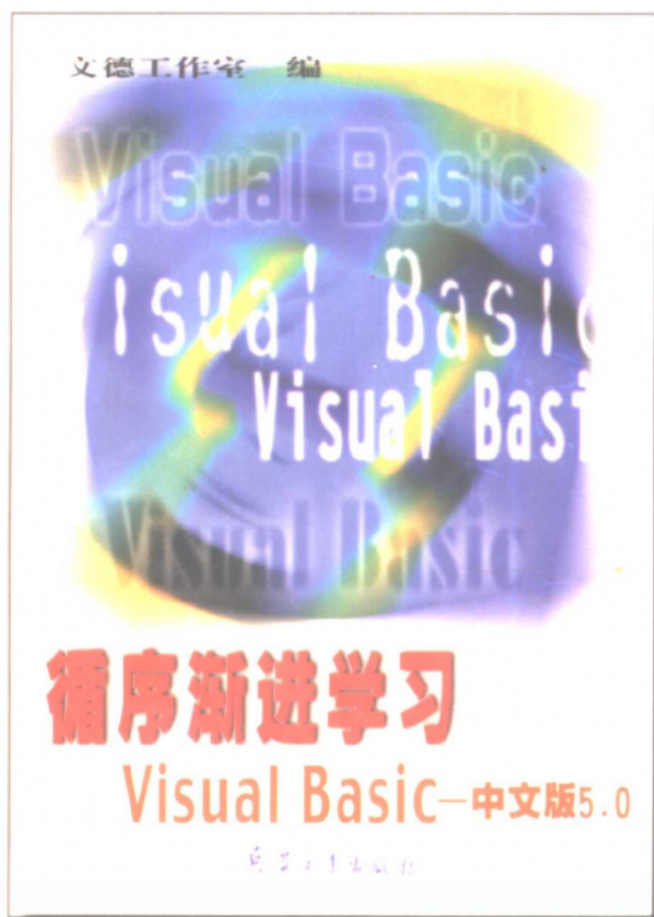
文德工作室 编

循序渐进学习

Visual Basic—中文版5.0

兵器工业出版社

责任编辑：郭 佳
封面设计：底晓娟



ISBN 7-80132-607-5



9 787801 326072 >

ISBN 7-80132-607-5/TP·147

定价：48.00 元



循序渐进学习 Visual Basic ——中文版 5.0

文德工作室 编

兵器工业出版社

内容简介

Visual Basic 是 Windows 环境下出现的一种强有力的编程工具。利用 Visual Basic 来开发 Windows 应用程序和图形界面非常方便,非常直观。

本书结合了中文版的特点,由浅入深地详细介绍了 Visual Basic 5.0 的编程技术。前面五章属于 Visual Basic 的基础部分,内容包括 Visual Basic 5.0 入门、Visual Basic 编程基础、类与对象、窗体和标准控件。从第六章开始,涉及到 Visual Basic 深层次的开发与应用,包括多文档界面、工程管理、Visual Basic 与数据库、多媒体开发与应用、OLE 技术与服务器、开发 ActiveX 控件、Web 与 ActiveX 共七部分。

本书适合于学习和使用 Visual Basic 5.0 for Windows 的广大计算机爱好者,以及大专院校相关专业的师生和技术开发人员。

图书在版编目(CIP)数据

循序渐进学习 Visual Basic——中文版 5.0/文德工作室
编. - 北京:兵器工业出版社,1999.5
ISBN 7-80132-607-5

I. 循… II. 文… III. Basic 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (1999) 第 10520 号

出版发行:兵器工业出版社

责任编辑:郭佳

责任技编:魏丽华

社址:100089 北京市海淀区车道沟 10 号

经销:各地新华书店

印刷:北京黄坎印刷厂印装

版次:1999 年 5 月第 1 版第 1 次印刷

印数:1—3000

封面设计:底晓娟

责任校对:冯敏悦 王 锋

责任印制:王京华

开本:787×1092 1/16

印张:31

字数:767.52 千字

定价:48.00 元

(版权所有 翻印必究 印装有误 负责调换)

前 言

Visual Basic 是 Microsoft 公司于 1991 年推出的一种程序设计语言。它综合运用了 Basic 语言和可视化的设计工具,并采用了事件驱动编程机制,因而非常适用于图形界面的编程方式。

Visual Basic 以丰富的用户图形界面(GUI)及其强有力的集中开发环境(IDE),为 Windows 应用程序,特别是图形用户界面应用程序的开发提供了一个有效的工具,从而使复杂的 Windows 编程工作变得十分简单便捷。

本书系统地介绍了中文版 Visual Basic 5.0 程序设计语言。首先详细地介绍了 Visual Basic 5.0 的基础知识,以便读者能够尽快地掌握 Visual Basic 的入门知识。接下来结合实际编程深入浅出地讲述了如何利用 Visual Basic 5.0 进行应用程序的开发,比如多媒体开发、数据库程序开发、定制 ActiveX 控件、创建 Internet 部件下载包等。

全书的介绍由浅入深、图文并茂、内容全面、论述深入。文中例举了大量的示例应用程序,能够为程序员提供有效的帮助。

本书的执笔者为杨志波和周后权。文德工作室的詹文军、张宇宏、廖铮等其他同志提供了大量的帮助,在此深表谢意。此外,还要感谢黄志伟、曹阳、肖俊东、李洋、刘欣、魏瑗等对本书工作的大力支持。熊奇峰同志对本书的编审作了大量的工作,并提供了许多宝贵的意见,这里一并致谢。

由于作者水平有限,加上成书时间仓促,本书中错误和疏漏之处在所难免,敬请读者批评指正。

作者

1998 年 10 月

目 录

第一章 Visual Basic 5.0 入门	(1)
1.1 概述	(1)
1.1.1 Visual Basic 的版本简介	(1)
1.1.2 事件驱动编程	(2)
1.1.3 Visual Basic 代码编写的基础知识	(3)
1.1.4 获得帮助	(4)
1.1.5 Visual Basic 5.0 的安装	(5)
1.2 工作窗口介绍	(8)
1.2.1 窗体	(9)
1.2.2 工具栏	(9)
1.2.3 数字显示区	(10)
1.2.4 工具箱	(10)
1.2.5 属性窗口	(11)
1.2.6 工程窗口	(12)
1.2.7 窗体布局窗口	(12)
1.3 工作模式	(13)
1.3.1 设计模式	(13)
1.3.2 运行模式	(14)
1.3.3 中断模式	(14)
1.4 定制工作环境	(15)
1.4.1 编辑器选项卡	(15)
1.4.2 通用选项卡	(18)
1.5 Visual Basic 的调试手段	(19)
1.5.1 交互式开发语言	(19)
1.5.2 错误类型	(20)
1.5.3 调试工具	(21)
第二章 Visual Basic 编程基础	(29)
2.1 变量	(29)
2.1.1 声明变量	(30)
2.1.2 变量范围	(35)

2.1.3 变量生存期	(39)
2.2 常量	(40)
2.3 数据类型	(42)
2.3.1 Numeric(数值型)数据类型	(42)
2.3.2 Byte(字节)数据类型	(43)
2.3.3 String(字符串)数据类型	(43)
2.3.4 Boolean(布尔)数据类型	(44)
2.3.5 Date(日期)数据类型	(45)
2.3.6 Object(对象)数据类型	(46)
2.3.7 Variant(变体)数据类型	(46)
2.3.8 转换数据类型	(47)
2.3.9 自定义数据类型	(48)
2.4 过程	(49)
2.4.1 子程序和函数	(49)
2.4.2 参数传递机制	(52)
2.4.3 使用可选参数	(54)
2.5 流程控制	(56)
2.5.1 条件语句	(56)
2.5.2 循环语句	(59)
2.5.3 嵌套控制结构	(62)
2.5.4 退出控制结构和过程	(63)
第三章 类与对象	(65)
3.1 第一个程序	(65)
3.1.1 建立控件对象	(65)
3.1.2 编写程序代码	(68)
3.1.3 运行程序	(69)
3.2 Visual Basic 的事件驱动机制	(70)
3.2.1 事件驱动机制	(70)
3.2.2 Visual Basic 的常用事件	(71)
3.3 对象的使用	(74)
3.3.1 基本概念	(74)
3.3.2 Visual Basic 的对象及其使用	(76)
3.4 使用类模块	(84)
3.4.1 类模块与其它模块	(84)
3.4.2 创建自己的类	(86)
3.5 使用对象浏览器	(92)
3.5.1 浏览和查看对象	(92)
3.5.2 改变默认属性	(93)

第四章 窗体	(95)
4.1 窗体概述	(95)
4.1.1 窗体的属性、方法和事件	(95)
4.1.2 应用程序的启动与退出	(103)
4.1.3 窗体的生存期	(106)
4.2 菜单的设计	(112)
4.2.1 创建菜单控件	(113)
4.2.2 为菜单项编写代码	(114)
4.3 工具栏的设计	(116)
4.3.1 创建工具栏	(116)
4.3.2 为工具栏控件编写代码	(119)
4.4 使用应用程序向导	(120)
4.4.1 应用程序向导的使用	(121)
4.4.2 添加控件和代码	(122)
4.4.3 单文档界面	(124)
第五章 标准控件	(126)
5.1 Visual Basic 控件简介	(126)
5.1.1 控件分类	(126)
5.1.2 内部控件简介	(127)
5.1.3 添加或删除 ActiveX 控件	(128)
5.1.4 控件命名约定	(128)
5.2 Label 控件	(130)
5.3 TextBox 控件	(133)
5.3.1 TextBox 控件的属性和事件	(133)
5.3.2 示例:口令应用程序	(138)
5.4 CommandButton 控件	(140)
5.4.1 CommandButton 控件的属性和事件	(140)
5.4.2 控件数组	(142)
5.4.3 示例:电话机应用程序	(143)
5.5 OptionButton 与 CheckBox 控件	(146)
5.5.1 OptionButton 控件	(146)
5.5.2 CheckBox 控件	(147)
5.6 Frame 控件	(148)
5.6.1 Frame 控件的属性和事件	(148)
5.6.2 示例:简单的计数器应用程序	(149)
5.7 PictureBox 与 Image 控件	(152)
5.7.1 PictureBox 控件	(152)

5.7.2 Image 控件	(159)
5.8 Timer 控件	(159)
5.8.1 Timer 控件的属性和事件	(160)
5.8.2 示例:计时器的应用	(160)
5.9 Shape 与 Line 控件	(161)
5.9.1 Shape 控件	(161)
5.9.2 Line 控件	(161)
5.10 HScrollBar 与 VScrollBar 控件	(162)
5.10.1 HScrollBar 与 VScrollBar 控件的属性和事件	(162)
5.10.2 示例:滚动条的应用	(163)
5.11 ListBox 与 ComboBox 控件	(164)
5.11.1 ListBox 控件	(165)
5.11.2 ComboBox 控件	(167)
5.11.3 示例:列表框和组合框的应用	(168)
5.12 文件系统控件	(170)
5.12.1 DriveListBox 控件	(170)
5.12.2 DirListBox 控件	(172)
5.12.3 FileListBox 控件	(173)
5.12.4 示例:文件浏览器应用程序	(174)
5.13 CommonDialog 控件	(177)
5.13.1 “打开”和“另存为”对话框	(178)
5.13.2 “颜色”对话框	(179)
5.13.3 “字体”对话框	(180)
5.13.4 “打印”对话框	(181)
5.13.5 显示帮助文件	(183)
第六章 多文档界面	(184)
6.1 创建 MDI 应用程序	(184)
6.1.1 手动创建 MDI 应用程序	(184)
6.1.2 自动创建 MDI 应用程序	(187)
6.1.3 为 MDI 应用程序添加控件和代码	(188)
6.2 创建对话框	(190)
6.2.1 模式与无模式对话框	(190)
6.2.2 预定义对话框	(191)
6.2.3 用窗体作为自定义对话框	(192)
6.2.4 关于对话框	(197)
6.3 文件的操作	(198)
6.3.1 新建	(198)
6.3.2 打开	(199)

6.3.3 保存	(202)
6.3.4 关闭	(203)
6.3.5 打印	(204)
6.4 MDI 应用程序示例	(205)
6.4.1 RichTextBox 控件	(206)
6.4.2 文字处理应用程序界面介绍	(213)
6.4.3 文字处理应用程序清单	(217)
第七章 工程管理	(236)
7.1 工程文件的构成	(236)
7.1.1 工程文件	(236)
7.1.2 工程文件的构成	(237)
7.2 工程中的文件和对象操作	(243)
7.2.1 文件操作	(243)
7.2.2 对象操作	(244)
7.2.3 使用外接程序	(246)
7.3 发行执行文件	(247)
7.4 创建安装程序	(250)
7.4.1 发行程序	(251)
7.4.2 创建并测试安装程序	(251)
第八章 Visual Basic 与数据库	(255)
8.1 关系数据库简介	(255)
8.2 Data 控件及其应用	(257)
8.2.1 Data 控件	(257)
8.2.2 示例:数据库查询	(265)
8.3 使用绑定控件	(269)
8.3.1 DBListBox 和 DBComboBox 控件	(270)
8.3.2 DBGrid 控件	(271)
8.3.3 MSFlexGrid 控件	(272)
8.4 示例:小型数据库开发系统	(272)
8.4.1 设置有关属性	(272)
8.4.2 编写程序代码	(274)
8.4.3 运行示例	(280)
8.5 SQL 简介	(283)
8.5.1 SQL 语句	(283)
8.5.2 示例:SQLTEST 应用程序	(287)
8.5.3 SQL 中的连接	(289)

8.6 Visual Basic 中的数据库工具	(290)
8.6.1 使用可视化数据管理器	(290)
8.6.2 数据窗体向导	(292)
第九章 多媒体开发与应用	(293)
9.1 多媒体基本知识	(293)
9.1.1 图像基本知识	(293)
9.1.2 数字音频基本知识	(296)
9.2 使用 MCI 控件	(298)
9.2.1 MCI 控件描述	(298)
9.2.2 MCI 控件的属性与事件	(299)
9.2.3 示例:AVIPlayer 应用程序	(302)
9.3 MCI 函数	(307)
9.3.1 命令字符串接口的 MCI 函数	(309)
9.3.2 示例:MCITEST 应用程序	(311)
9.3.3 命令消息接口的 MCI 函数	(313)
9.4 使用 MCI 指令	(316)
9.4.1 多媒体设备	(316)
9.4.2 MCI 指令初步	(317)
9.4.3 系统指令	(318)
9.4.4 需求指令	(319)
9.4.5 基本指令	(321)
9.4.6 加强指令	(323)
9.4.7 示例:录音机	(323)
9.5 调用 API 函数	(329)
9.5.1 使用 API 浏览器	(330)
9.5.2 多媒体 API 函数的应用	(331)
9.5.3 几个有用的 API 函数	(337)
9.6 动画制作	(337)
9.6.1 利用 MOVE 方法	(338)
9.6.2 利用 LoadPicture 方法	(339)
9.6.3 调色板动画	(341)
9.6.4 使用 BitBlt() 函数	(347)
9.7 超媒体入门	(352)
9.7.1 基本概念	(352)
9.7.2 示例:帮助文件	(353)
第十章 OLE 技术与服务器	(358)

10.1 OLE 技术	(358)
10.1.1 几个概念	(359)
10.1.2 设计时的对象链接与嵌入	(361)
10.1.3 运行时的对象链接与嵌入	(367)
10.2 Excel 服务器	(371)
10.2.1 Microsoft Excel 的对象模型	(372)
10.2.2 创建对象	(372)
10.2.3 使用 Excel 服务器中的方法和属性	(373)
10.2.4 OLE 自动化	(375)
10.3 编写服务器程序	(377)
10.3.1 服务器执行的地点	(377)
10.3.2 Instancing 属性	(378)
10.3.3 向系统注册	(379)
10.3.4 编写服务器程序	(379)
10.3.5 编写客户应用程序	(380)
10.3.6 示例:录音机服务器	(381)
第十一章 开发 ActiveX 控件	(383)
11.1 ActiveX 控件界面向导的使用	(383)
11.1.1 六个步骤	(384)
11.1.2 ActiveX 控件的三种方式及其框架	(389)
11.2 ActiveX 控件的属性、方法和事件	(398)
11.2.1 ActiveX 控件的属性	(399)
11.2.2 ActiveX 控件的方法	(404)
11.2.3 ActiveX 控件的事件	(407)
11.2.4 属性页设计	(408)
11.3 完善 Rotating 控件	(416)
11.4 Rotating 控件的应用	(416)
第十二章 Web 与 ActiveX	(420)
12.1 Web 文档与 HTML 语言	(420)
12.1.1 HTML 介绍	(421)
12.1.2 HTML 的常用标签	(422)
12.1.3 HTML 的多媒体标签	(424)
12.2 Visual Basic 脚本编写	(428)
12.2.1 VBScript 简介	(429)
12.2.2 如何编写 VBScript	(429)
12.2.3 标准 HTML 控件	(431)
12.2.4 使用属性和方法	(436)

12.2.5 创建过程·····	(438)
12.3 下载 ActiveX 控件 ·····	(440)
12.3.1 概述·····	(440)
12.3.2 下载 ActiveX 控件的步骤 ·····	(441)
12.3.3 给 HTML 页添加控件 ·····	(445)
12.3.4 对控件进行许可证管理·····	(447)
12.4 VBScript 对象与编程 ·····	(450)
12.4.1 VBScript 模型的对象 ·····	(450)
12.4.2 对象的属性和方法·····	(451)
12.4.3 利用 VBScript 为对象编程 ·····	(455)
12.5 WebBrowser 控件和 InternetExplorer 对象 ·····	(460)
12.5.1 WebBrowser 控件简介 ·····	(460)
12.5.2 WebBrowser 控件的应用 ·····	(460)
12.5.3 InternetExplorer 对象简介 ·····	(468)
12.5.4 InternetExplorer 对象的应用 ·····	(469)
12.5.5 在 Visual Basic 应用程序中使用 WebBrowser 控件和 InternetExplorer 对象 ·····	(471)
12.6 ActiveX 文档 ·····	(472)
12.6.1 ActiveX 文档简介 ·····	(472)
12.6.2 创建 ActiveX 文档工程 ·····	(474)
12.6.3 测试和调试 ActiveX 文档 ·····	(476)
12.6.4 多文档工程·····	(478)
12.6.5 扩展用户界面·····	(480)

第一章 Visual Basic 5.0 入门

Visual Basic(简称 VB)以其独特的简单便捷的编程方式开发图形用户界面(GUI),从而使编程人员可以轻松自如地开发 Windows 应用程序,而不再需要编写大量代码去描述界面元素的外观和位置,只要把预先建立的对象拖放到 Visual Basic 特有的窗体(Form)上即可。这样就可以节省大量的编程时间和繁杂的调试工作,为广大用户带来极大的方便。

本章主要介绍 Visual Basic 5.0 的入门知识。首先简单介绍了 Visual Basic 语言的一些基本情况,比如版本、安装等。接下来介绍了 Visual Basic 5.0 的集成开发环境和工作模式,还介绍了 Visual Basic 代码编写的基础知识及其提供的调试手段,以便读者一开始便对 Visual Basic 5.0 有个大概的轮廓印象。

本章的内容包括以下几部分:

- ◆ Visual Basic 概述
- ◆ Visual Basic 的工作窗口
- ◆ Visual Basic 的工作模式
- ◆ 如何定制 Visual Basic 的工作环境
- ◆ Visual Basic 提供的调试手段

1.1 概 述

本节介绍 Visual Basic 的一些基础知识,包括 Visual Basic 的版本简介、Visual Basic 的事件驱动编程、Visual Basic 代码编写的基础知识、如何获得帮助,以及 Visual Basic 5.0 的安装。

1.1.1 Visual Basic 的版本简介

Visual Basic 提供了三个版本,以满足不同层次的开发需要。

● 学习版

Visual Basic 学习版是最基础的入门版本,主要用来开发 Windows 95 和 Windows NT 的应用程序。该版本包括所有的内部控件,以及网格、图表和数据绑定控件等。

● 专业版

Visual Basic 专业版是专门为专业编程人员提供的一套进行 Windows 程序开发的功能完备的工具集。该版本不仅包括学习版的全部功能,还包括 ActiveX 控件、Internet 控件和 Visual Basic Crystal 报告等高级功能。

● 企业版

Visual Basic 企业版是最高级的版本,专业编程人员可以利用其来开发功能强大的分布式

应用程序。该版本除包括专业版的全部功能外,还包括自动化管理器(Automation Manager)、部件管理器(Component Manager)、数据库管理工具(Database Manage Tools)以及面向工程版本的 Microsoft Visual SourceSafe 等。

1.1.2 事件驱动编程

Visual Basic 起源于 Basic 语言这一结构化编程语言。但是,Visual Basic 使用事件驱动编程模型。

在传统的或过程式的应用程序中,应用程序自身控制着代码的运行,程序按照预先确定的路径顺序执行,只是在必要时进入循环或调用过程。程序的执行是一种结构化的进程。

在事件驱动的应用程序中,代码并不是按照预定的路径顺序执行,而是根据不同的事件作出响应,从而运行响应的代码段。事件可以由用户触发,也可由来自操作系统或其它应用程序的消息触发,甚至可以由该应用程序自身的消息触发。事件顺序取代了代码执行的顺序。因此,每次运行程序,代码根据事件所作出的响应的不同,其所执行的路径也是不同的。

虽然事件得到触发的顺序没法事先预测,但事件还是存在特定的类型的。比如命令按钮的事件一般包括单击(Click)事件、双击(DblClick)事件;而文本框则包括文本框中文本改变的 Change 事件,以及输入文本时对按键响应的 KeyPress 事件等。当单击命令按钮时,Click 事件过程得到触发,程序便去执行 Click 事件过程的代码段,而不管其在整个程序代码所处的位置。一旦在文本框中输入字符,立即触发 Change 事件,其相应的事件过程代码段得到执行。但是,这种触发的顺序是无法预知的。

事件在得到触发时,总是去执行其代码段。但如果事件过程没有代码,是没法得到响应的。因此,代码的编写就关系到整个应用程序的成败。编写代码是事件驱动编程的核心部分,这些代码应该对特定对象指定可能发生的事件过程,以便对事件作出响应。比如前面提到的按钮的 Click 事件和文本框的 Change 事件。

事件驱动模型除了事件外,还有一个必不可少的元素——对象。Visual Basic 是一种面向对象的语言。所有的事件过程都是由对象而得到触发的。当对象捕捉到触发消息时,便执行该对象相应的事件过程。比如单击某一按钮 Command1,便触发一个 Click 事件,但得到响应的过程却是由单击的对象 Command1 决定的,这一得到触发的事件过程可描述为 Command1-Click()。可以看出,事件过程是 Visual Basic 程序代码得到执行的驱动机制。事件过程由对象名、下划线和事件名组成。事件驱动机制更为详细的情况可参见第三章。

在 Visual Basic 中,对象是一个很广泛的概念。比如最顶级的对象是 Window 对象,而前面提到的按钮则是 Visual Basic 控件对象中的一个。Visual Basic 的控件(Control)对象是其最基本的对象,而控件对象又包括内置式控件、ActiveX 控件和嵌入式控件。Visual Basic 还有一个基本的对象是窗体(Form)。利用窗体和各种控件对象,创建 Visual Basic 的图形用户界面,再加上响应对象事件过程的代码,便构成了 Visual Basic 的应用程序。因此 Visual Basic 应用程序的开发由下面两个基本步骤组成:

1. 创建应用程序的用户界面;
 2. 编写对用户界面中对象的行为作出响应的代码。
- 如何利用对象进行编程的有关知识可参见第三章。

1.1.3 Visual Basic 代码编写的基础知识

在开始设计 Visual Basic 的应用程序之前,首先应该掌握 Visual Basic 代码的一些书写规则。

将单行语句分成多行

Visual Basic 的代码以回车作为一个代码行的结束,不用像 C 语言那样要用分号(;)作为终结符。如果代码语句比较长,一行写不下,可以分成多行书写。Visual Basic 采用的续行符为空格加下划线(-)。使用续行符将长语句变成多行代码,不仅便于阅读,而且与打印时的效果一样(因为打印时会强制换行)。下面的例子便采用了续行符,将 MsgBox 函数语句分成了两行:

```
MsgBox "Hello, world" -  
    , vbOKCancel, "Hello"
```

但是在续行符后边是不能加注释行的。比如下列代码是错误的,Visual Basic 将给出编译错误消息。

```
MsgBox "Hello, world" -      ' 输出消息框  
    , vbOKCancel, "Hello"
```

将多个语句合并写在一行

可以在一个代码行写多条语句,特别是比较短的语句。将两个或两个以上的语句放在同一行时,要用冒号(:)将它们分开。比如可以在一行进行多个变量的声明和赋值:

```
Dim a As Integer: Dim y As Boolean: a = 10: y = True
```

上述代码虽然没有错误,但不便于阅读。因此最好还是一行写一个语句。

在代码中添加注释

单纯的代码是很难读懂的,特别是读别人编写的程序。因此在编写代码时,有必要加上适当的语句进行描述和说明。Visual Basic 采用的注释符为单引号('),单引号右边的内容是代码的注释。Visual Basic 的编译器遇到单引号时,将其右边的语句行当作注释行忽略,并不对其进行编译。比如:

```
Private Sub Command1-Click()  
    ' 输出消息框  
    MsgBox "Hello, world", vbOKCancel, "Hello"  
    Dim a As Integer      ' 声明变量  
    a = 10                ' 变量初始化  
End Sub
```

从上可以看出,注释行可以写在与代码同一行的右边,也可以单独写一行。但注释行右边不能再存在有效代码,因为 Visual Basic 把单引号右边的所有字符均当作注释忽略了。还应该注意一点,注释行不能出现在与续行符同一行的右边。

缩进

利用缩进可以区分代码的不同部分,例如循环和条件语句。上面代码便采用了缩进。要缩进所选的代码,可使用 Tab 键或【编辑】菜单中的【缩进】命令。类似的,使用 Shift + Tab 键或【编辑】菜单中的【凸出】命令,可以使选定的代码向左移动一个制表位。

1.1.4 获得帮助

Visual Basic 提供了丰富的帮助,开发人员既可从联机手册中查阅有关信息,又可在 Visual Basic 开发环境中动态地获得联机帮助。如果已连接在 Internet 网上,还可以访问 Microsoft 的相关站点以获得技术支持,从而为开发人员带来了极大的便利。

联机手册

Visual Basic 在安装时已将联机手册安装到机器上了,可单击【开始】、【程序】、Microsoft Visual Basic 5.0 中的【联机手册】,以启动联机手册应用程序。也可在 Visual Basic 编程环境下选择【帮助】菜单中的【联机手册】命令来启动联机手册程序。

联机手册提供了最为广泛的帮助信息来源,其中包括 Visual Basic 5.0 的新内容、程序员指南、部件工具指南、数据访问指南等等,如图 1.1 所示。

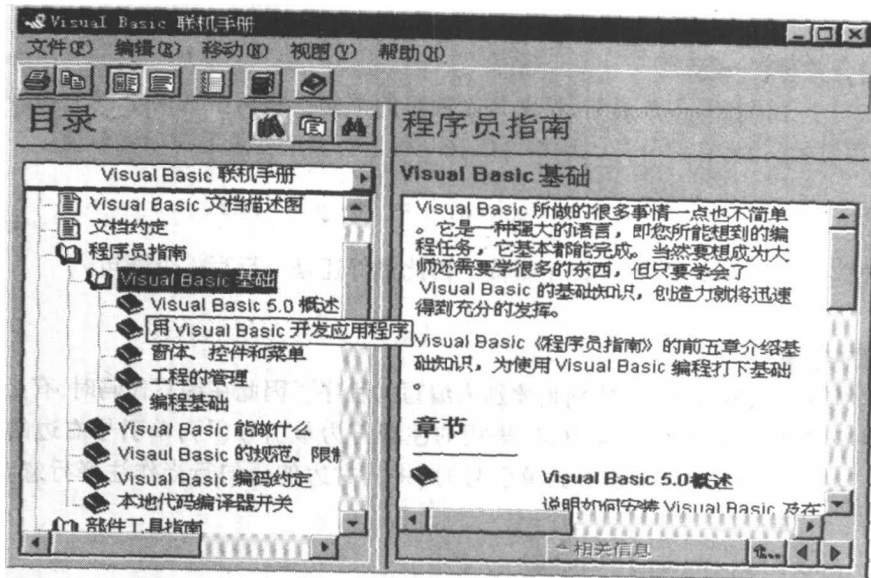


图 1.1 Visual Basic 联机手册

联机手册包括了 Visual Basic 的所有文档,并提供了几种查找信息的方法,可以很方便地找到相关的信息主题。联机手册的查找方法有:使用目录表、使用关键词、使用全文搜索等。

联机帮助

Visual Basic 提供了广泛的联机帮助,并在帮助文件中提供了许多代码示例,这些代码可以直接复制到用户的应用程序中来。在 Visual Basic 编程环境下单击【帮助】菜单中的【Mi-