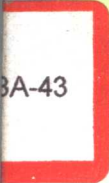


管曙亮 姜大庆 李鹏祥 编



高等教育出版社

高等职业学校教材

Visual Basic 应用基础

管曙亮 姜大庆 李鹏祥 编

 高等教育出版社

内容提要

本书详细讲述了 Visual Basic 6.0 的基本内容,列举了大量的应用实例,使读者在短期内迅速地步入 Windows 应用程序开发领域,了解到 Visual Basic 易学易用、功能强大的特点。本书从基本概念、基本操作到多媒体、数据库应用程序开发等做了较详细的讲解,在程序开发中有较高的参考价值。本书分为基础篇和应用篇两个层次,结构组织合理,便于教学取舍。针对各章重点,精心编写习题,以巩固所学内容。

本书可作为高等职业学校计算机应用专业的教材,亦可供中等职业学校选用,也是 Visual Basic 初学者理想的入门读物,对计算机工作者及爱好者也有很好的参考价值。

图书在版编目(CIP)数据

Visual Basic 应用基础/管曙亮等编. —北京:高等教育出版社,2001.7

ISBN 7-04-009238-7

I. V… II. 管… III. BASIC 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2000)第 86201 号

Visual Basic 应用基础

管曙亮 姜大庆 李鹏祥 编

出版发行 高等教育出版社

社 址 北京市东城区沙滩后街 55 号

电 话 010—64054588

网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>

邮政编码 100009

传 真 010—64014048

经 销 新华书店北京发行所

印 刷 北京民族印刷厂

开 本 787 × 1092 1/16

印 张 19.5

字 数 460 000

版 次 2001 年 7 月第 1 版

印 次 2001 年 7 月第 1 次印刷

定 价 26.50 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

前 言

Visual Basic(简称 VB)继承了原有 Basic 语言简单易学的优点,采用可视化(Visual)、面向对象、事件驱动的程序设计模式,大大简化了 Windows 应用程序的设计,从而成为目前 Windows 应用程序的最便捷、最有效的开发工具之一。许多学校已经开设或即将开设 VB 这门课程。

本书可作为高职学校计算机应用及相关专业的教材,又可作为各类相关培训班的培训教材。本书不仅让学生了解面向对象程序设计的概念、掌握可视化开发工具的使用,还注重对学生实际操作能力的培养,既让学生了解 VB 能做什么,又使学生掌握如何去做,在注重教材系统性、科学性的同时,突出实用性和操作性,通过详尽的基本知识叙述和大量的应用实例,激发他们的学习热情,使学生学后会用,学以致用。

本教材分为基础篇和应用篇两部分,便于各校安排教学。

基础篇共分八章,是学习 VB 必须掌握的内容。第一章介绍面向对象程序设计的特点及基本概念,为本课程的学习奠定坚实的基础;第二章介绍 VB 集成开发环境的使用和应用程序开发步骤;第三章介绍 VB 程序设计必备的基本知识;第四章、第五章分别对输入、输出的各种手段作了详略得当的介绍,使学生能在比较中学得透、用得活;第六章在前两章的基础上进一步介绍应用程序界面设计;第七章介绍用于文件管理与读写的各种命令,使用随机文件处理简单数据;第八章作为基础篇的结束,介绍了如何查找错误和进行出错处理。

应用篇是使用 VB 开发实用程序的选学内容。第九章介绍多媒体控件使用方法,并列举了大量多媒体程序实例,有较高参考价值;第十章介绍数据库建立、记录集操作、SQL 数据查询等基本内容,加强了 ADO 对象、数据环境对象、数据报表设计新知识的叙述,还介绍了使用“应用程序向导”迅速生成数据库应用程序的过程,实用性很强。

本教材的编者都是从事 Basic 或 VB 教学多年的教师,很了解教学对象的知识水平、接受能力和目标要求,熟知学生学习中可能出现的问题,因而教材的编写有极强的针对性。行文朴实流畅、浅显易懂,图文并茂,示例丰富,便于自学。每章有相应的概念题、操作题,对巩固读者所学内容有很好的帮助作用。

本书由管曙亮任主编。其中,第一章、第二章、第六章(前四节)、第九章和附录由姜大庆执笔;第三章和第七章由李鹏祥执笔;第四章、第五章、第六章(第五节)、第八章和第十章由管曙亮执笔。江苏省淮海工学院的陆卫中老师对全稿进行了认真的审阅,并提出了宝贵的修改意见,在此表示衷心感谢。

由于作者水平有限，书中肯定存在不足之处，敬请读者批评指正。

本教材计划教学时数为 80 学时，具体教学时数的分配可参考下表。

内 容	教学时数	内 容	教学时数
第一章	2	第六章	12
第二章	4	第七章	8
第三章	10	第八章	4
第四章	12	第九章	6
第五章	8	第十章	14
合计	80		

以上学时包括授课与实验，在课时充足的情况下可增加实验学时或增加一周的课程设计。

编 者

2000 年 9 月

目 录

第 1 章 Visual Basic 概述	1
1.1 VB 发展简介	1
1.2 VB 的编程特点	2
1.3 VB 程序设计中的基本概念	4
1.3.1 对象	4
1.3.2 属性	5
1.3.3 事件	6
1.3.4 方法	6
习题一	7
第 2 章 VB 集成开发环境	8
2.1 VB 6.0 集成开发环境中的元素	8
2.1.1 标题栏	9
2.1.2 菜单栏	9
2.1.3 工具栏	10
2.1.4 工具箱	11
2.1.5 窗体设计器	11
2.1.6 窗体布局窗口	12
2.1.7 代码窗口	12
2.1.8 工程资源管理器窗口	13
2.1.9 属性窗口	13
2.2 用 VB 开发应用程序	14
2.2.1 VB 应用程序的结构	14
2.2.2 应用程序开发步骤	15
2.3 创建第一个 VB 应用程序	15
2.3.1 创建一个新的工程	16
2.3.2 创建用户界面	16
2.3.3 设置窗体和控件的属性	17
2.3.4 编写对象的事件过程代码	19

2.3.5 运行、调试程序	20
2.3.6 工程的保存和打开	20
2.3.7 建立可执行文件	21
习题二	21
第3章 VB 程序设计基础	24
3.1 VB 与 Basic 语言	24
3.1.1 标识符	24
3.1.2 注释语句	25
3.1.3 语句分隔符和续行符	25
3.1.4 赋值语句	26
3.2 VB 数据类型	27
3.2.1 布尔型 (Boolean)	27
3.2.2 数值型	27
3.2.3 字符串型 (String)	29
3.2.4 日期型 (Date)	30
3.2.5 变体型 (Variant)	30
3.2.6 数据类型转换	31
3.2.7 对象型 (Object)	32
3.2.8 自定义类型	32
3.3 常量、变量、数组和集合	33
3.3.1 常量与符号常量	33
3.3.2 变量的定义与变量的作用域	34
3.3.3 数组的定义与使用	36
3.3.4 集合对象的创建	37
3.3.5 集合对象的属性和方法	38
3.3.6 集合与数组的区别	40
3.4 运算符与表达式	41
3.4.1 算术运算符与算术表达式	41
3.4.2 关系运算符与关系表达式	42
3.4.3 逻辑运算符与逻辑表达式	42
3.4.4 字符运算符与字符表达式	42
3.4.5 各类运算符的运算优先级	43
3.5 程序控制结构	43
3.5.1 选择结构	43
3.5.2 循环结构	46
3.5.3 程序控制结构的综合使用	50
3.6 过程	52

3.6.1 常用的标准函数	52
3.6.2 子过程 (Sub 过程) 的定义	53
3.6.3 函数过程 (Function 过程) 的定义	55
3.6.4 过程调用	56
3.6.5 过程的参数传递	57
习题三	59
第 4 章 输入	61
4.1 控件概述	61
4.1.1 控件的分类	61
4.1.2 在窗体上添加控件	62
4.2 用于输入的控件	63
4.2.1 TextBox 控件	63
4.2.2 CommandButton 控件	75
4.2.3 CheckBox 控件、OptionButton 控件与 Frame 控件	78
4.2.4 ListBox 控件与 ComboBox 控件	82
4.2.5 HScrollBar 控件与 VScrollBar 控件	86
4.3 InputBox 函数	88
4.4 CommonDialog 控件	89
4.4.1 通用对话框的特点	89
4.4.2 通用对话框的主要属性	89
4.4.3 与文件对话框有关的属性	90
4.4.4 与颜色对话框有关的属性	91
4.4.5 与字体对话框有关的属性	91
4.4.6 与打印对话框有关的属性	91
4.5 RichTextBox 控件	93
习题四	96
第 5 章 输出	99
5.1 用于输出的基本控件	99
5.1.1 Label 控件	99
5.1.2 PictureBox 控件与 Image 控件	103
5.1.3 Line 控件与 Shape 控件	106
5.2 VB 绘图	111
5.2.1 绘图区当前位置属性	111
5.2.2 VB 绘图方法	112
5.2.3 VB 动画设计与 Timer 控件	117
5.3 MsgBox 函数	121

5.4 打印输出	124
5.4.1 使用 Printer 对象直接打印	124
5.4.2 使用窗体对象间接打印	125
习题五	125
第 6 章 界面设计	127
6.1 窗体的主要属性、方法和事件	127
6.1.1 窗体的属性	128
6.1.2 窗体的部分事件	129
6.1.3 窗体的部分方法	131
6.2 多重窗体程序设计与自定义对话框	132
6.2.1 多重窗体程序设计	132
6.2.2 创建自定义对话框	134
6.3 多文档界面设计	140
6.3.1 MDI 父窗体与 MDI 子窗体	140
6.3.2 建立 MDI 应用程序	142
6.3.3 MDI 的主要属性、方法、事件	143
6.4 菜单设计	146
6.4.1 下拉式菜单设计	146
6.4.2 菜单项的控制	150
6.4.3 弹出式菜单设计	153
6.5 优化用户界面	155
6.5.1 创建工具栏 (ToolBar)	156
6.5.2 创建状态栏 (StatusBar)	159
6.5.3 创建选项卡 (TabStrip)	162
习题六	164
第 7 章 文件管理	168
7.1 用于文件管理的语句和函数	168
7.1.1 获取当前目录(CurDir 函数)	168
7.1.2 改变当前目录(ChDir 语句)	169
7.1.3 建立和删除目录(Mkdir 语句和 Rmdir 语句)	169
7.1.4 改变当前驱动器(ChDrive 语句)	169
7.1.5 复制和删除文件(FileCopy 语句和 Kill 语句)	170
7.1.6 设置文件的属性(SetAttr 语句)	170
7.1.7 文件改名(Name 语句)	171
7.1.8 获取文件名或目录名(Dir 函数)	171
7.1.9 打开或执行一个文件(Shell 语句)	172

7.2 文件系统控件	174
7.2.1 DriveListBox 控件	174
7.2.2 DirListBox 控件	175
7.2.3 文件列表框(FileListBox)	177
7.2.4 协同文件系统控件	178
7.3 文件的读写	182
7.3.1 顺序文件	182
7.3.2 随机文件	185
7.3.3 使用随机文件管理数据库	188
7.3.4 二进制文件	193
习题七	194
第 8 章 程序调试与出错处理	196
8.1 VB 应用程序中的错误类型	196
8.1.1 编译错误	196
8.1.2 运行时错误	196
8.1.3 逻辑错误	197
8.2 使用调试工具发现逻辑错误	197
8.2.1 设计时、运行时和中断模式	198
8.2.2 进入中断模式	198
8.2.3 跟踪应用程序的执行	200
8.3 设置错误陷阱与出错处理	206
8.3.1 错误处理程序的结构	206
8.3.2 错误处理的分层结构	208
8.3.3 关闭错误处理	210
习题八	210
第 9 章 多媒体应用程序设计	211
9.1 MMControl 控件概述	211
9.2 MMControl 控件的常用属性	212
9.3 MMControl 控件的常用事件	218
9.4 VB6.0 多媒体程序设计实例	221
9.4.1 设计一个 AVI 动画播放器	222
9.4.2 设计一个 CD 播放器	224
习题九	227
第 10 章 数据库应用程序设计	229
10.1 了解数据库	230

10.1.1 数据库的概念	230
10.1.2 使用 VisData 建立 Access 数据库	231
10.2 ADO 对象与记录集操作	234
10.2.1 ADO 对象	234
10.2.2 记录集操作	242
10.3 ADO 数据控件与数据环境	249
10.3.1 ADO 数据控件	249
10.3.2 数据环境	253
10.4 数据绑定控件	257
10.4.1 DataList 控件与 DataCombo 控件	258
10.4.2 DataGrid 控件	259
10.4.3 MSHFlexGrid 控件	261
10.4.4 DTPicker 控件与 MonthView 控件	262
10.4.5 DataRepeater 控件	262
10.5 结构化查询语言 SQL	266
10.5.1 SQL 语句的生成器与运行方法	266
10.5.2 数据查询语言	271
10.6 数据库应用程序的建立	277
10.6.1 创建数据报表	277
10.6.2 创建数据库应用程序	283
习题十	285
附录一 VB6.0 常用函数	288
附录二 VB6.0 常用符号常数	291
附录三 VB6.0 常用关键字	297

第 1 章 Visual Basic 概述

【学习目标】 本章以 Visual Basic 6.0 中文版为背景介绍 Visual Basic 的基本概念及编程特点。通过本章的学习，读者应能对 Visual Basic 的发展和特点有总体的了解，并能叙述 Visual Basic 中对象、属性、事件和方法的基本概念及其简单应用。

Visual Basic (简称 VB) 程序设计语言是 Microsoft 公司于 1991 年推出的 Windows 应用程序开发工具。它继承了原有 Basic 语言的简单易学的优点，采用一种可视化 (Visual)、面向对象、事件驱动的程序设计模式，大大简化了 Windows 应用程序的设计，从而成为目前 Windows 应用程序的最便捷、最有效的开发工具之一。它的出现被计算机界称作是一个“令人震惊的奇迹”。

1.1 VB 发展简介

VB 的出现可以说是 Microsoft Windows 日渐成熟的必然产物。Windows 为程序员和最终用户提供了一个图形化的操作环境，为开发 Windows 下的应用程序，Windows 提供了一组预定义工具，即 Windows 的软件开发工具箱 (SDK)，该工具能使程序员建立一个具有 Windows 界面的应用程序。但是，SDK 提供的六百多个函数和与其一致的事件驱动编程技术，又使程序员重新陷入困境，程序员不仅要掌握事件驱动编程技术和六百多个函数功能，而且还得用 C 语言描述这些问题，这样，编程工作变得越来越复杂。因此程序员迫切需要一种操作方便，使用简单的开发工具来简化 Windows 应用程序的开发工作。在这样的背景下，新一代面向对象的开发工具——Visual Basic 由此诞生了。

VB 的发展也与 Windows 的发展有着紧密的联系。Microsoft 公司于 1991 年在传统 BASIC 语言基础上开发出 VB 1.0 版本，1992 年秋季推出 2.0 版，1993 年 4 月又推出了 3.0 版。3.0 版本以前的 VB 可在 Windows 3.x 环境下运行。随着 32 位的 Windows 95 操作系统的发行，Microsoft 公司于 1995 年秋季首次推出能开发 32 位应用程序的 VB 4.0 版本，这是 VB 发展史上的一次较大的飞跃。由于 Internet 的迅猛发展，Microsoft 公司于 1997 年推出了 VB 5.0 版，该版本增强了 VB 对 Internet 的支持能力，同时，VB 5.0 版首次引入了本机代码编译器，使开发出的应用程

序能真正编译成标准的 EXE 文件,大大提高了运行速度(同 VB 4.0 相比,执行速度提高了 30%),是 VB 发展史上又一次质的飞跃。1998 年秋季,随着 Windows98 的发行,Microsoft 公司推出了 VB 6.0 版,该版本在编制 Web 应用以及对数据库的访问功能等方面都得到了进一步的增强、丰富和提高。

从 1.0 到 4.0 版,VB 只有英文版,而 5.0 版以后的 VB 在推出英文版的同时,又推出了中文版,大大方便了中国用户,也为 VB 在中国软件行业的进一步推广运用奠定了良好的基础。

1.2 VB 的编程特点

对于用户来说,Windows 环境是相当出色的。它提供了一个直观的图形用户工作环境,用户通过对图标、菜单、工具栏、对话框等界面元素简单选择和操作就可以实现自己所需要的功能。但对编程人员来说,为了编写 Windows 下的应用程序,必须建立相应的窗口、菜单及对话框等“构件”,其工作的复杂程度反而增大了。VB 的出现为 Windows 下的编程提出了一个新的概念,它巧妙地将 Windows 编程的复杂性“封装”起来,综合运用 BASIC 语言和新的可视化设计工具,既没有牺牲 Windows 所特有的优良性能和图形工作环境,又提供了编程的简易性和高效性,可用来开发 Windows 环境下的各类应用程序。

总体来看,利用 VB 语言编程具有以下几个特点:

(1) 可视化程序设计

用传统的程序设计语言设计程序时,必须通过大量代码去描述界面元素的外观和位置,而且在设计过程中看不到界面的实际显示效果,必须编译后运行程序才能观察。一个成功的界面需经反复多次编程-编译-修改才能完成,这种编程方式会大大影响编程效率。VB 提供了可视化的设计工具,把 Windows 界面设计的复杂性“封装”起来。开发人员只需根据界面的设计要求,在 Windows 下建立一个“窗体”,并直接在窗体上画出各种“图形对象”,通过设置这些图形对象的“属性”,调整其在窗体界面中的位置、大小和样式,从而避免了为界面编写大量程序代码的工作,大大提高了程序设计的效率。

(2) 面向对象程序设计

VB 支持面向对象程序设计。在 VB 中,用来构成用户图形界面的可视化窗体及控件(如按钮、文本框、标签等)都是一个个对象。编程时用户可直接引用这些对象,并可以直接使用系统已经为对象封装好的各种功能,用户不必重新编写代码建立和描述每个对象,只需根据实际需要及每个对象所提供的功能编写程序即可。

(3) 事件驱动的程序设计

传统的编程是面向过程、按规定顺序进行的。应用程序执行时,从第一个语句开始执行,直到最后一个语句结束,这种程序结构完全由编程人员控制程序的执行。而 Windows 下的

应用程序,必须能让用户自由操纵程序的执行,需要采用事件驱动的程序设计方式完成。与 Windows 下的其他开发工具一样,VB 也采用事件驱动的编程机制,应用程序的执行是通过事件来驱动的。一个对象可能会产生多个事件,每个事件都可通过一段程序来响应。例如,命令按钮是一个常用的对象,当用户用鼠标在它上面单击一下时,便会在该对象上产生一个鼠标单击事件(Click),VB 会自动调用执行命令按钮上的 Click 事件过程,实现指定的操作。

在用 VB 设计应用程序时,没有明显的主程序概念,程序员所要做的工作是针对不同的对象分别填写它们相关的事件过程代码。因此,整个应用程序是由若干个这样过程程序组成的,从而降低了编程的难度和工作量,提高了程序开发效率。

(4) 强大的数据库编程能力

利用 VB 的数据控件和数据库管理器等工具,可直接建立或处理 Microsoft Access 格式的数据库,还能直接编辑和访问其他外部数据库,如 Dbase、Foxpro、Paradox 等。同时 VB 还提供开放式数据库访问(ODBC)功能,可通过直接访问或建立连接的方式使用并操作远程服务器上的关系型数据库,如 SQL Server、Oracle 等。使用结构化查询语言 SQL,轻松访问并操纵远程服务器上的关系型数据库,运用应用程序向导,迅速创建标准的数据库应用程序。

(5) 其他特性

① 支持动态链接库(DLL)。VB 是一种高级程序设计语言,不具备低级语言对机器硬件进行操作的功能,为此,VB 提供了访问动态链接库(Dynamic Linking Library)的功能,可以利用其他语言(如 C/C++等),将需要实现的功能编译成动态链接库,然后提供给 VB 调用。此外,通过动态链接库,还可调用功能强大的 Windows 应用程序接口(API)函数。API 是 Windows 环境中可供任何 Windows 应用程序访问和调用的一组函数集合。在 Windows 操作系统中,包含了 1 000 多个功能强大、经过严格测试的 API 函数,供程序员编程时直接调用。充分利用这些 API 函数,可大大增强 VB 的编程能力。

② 支持动态数据交换(DDE)。动态数据交换(Dynamic Data Exchange),就是把一种应用程序中的数据动态地链接到另一种应用程序中,当原始数据变化时,可以自动更新链接的数据。VB 提供了动态数据交换技术,可在应用程序中建立与其他 Windows 应用程序之间的动态数据交换的通道,使得应用程序在运行过程中可以相互交换信息。

③ 支持对象的链接与嵌入(OLE)。对象的链接与嵌入(Object Linking and Embedding),就是将每个应用程序都看做是一个对象,将多个不同的对象链接或嵌入到某个应用程序(它本身也是对象)中,如 WORD 文档中通过“插入/对象”可得到具有声音、图像、动画、文字、表格等各种形式的复合式文件,文件中每个对象都与创建它的应用程序相联系,双击这些对象即可在该应用程序中执行与创建这些对象的应用程序完全相同的操作。

④ 支持 Internet 应用程序的开发。在 VB 中还可以轻松地开发基于客户端的 DHTML 应用程序、基于服务器端的 IIS 应用程序、创建自己的 ActiveX 控件和在 Internet 浏览器上使用的 ActiveX 文档,这大大拓展了 VB 的 Internet 功能。

1.3 VB 程序设计中的基本概念

正如前面所述,传统的程序设计是面向过程的,这种程序的设计是模拟用户要解决问题的过程,围绕着函数或过程来进行,程序的执行也是顺序的;而 VB 程序设计则是面向对象的,应用程序是由对象事件驱动的,每个对象具有自己的状态和行为,即属性和方法。本节将就 VB 中的对象、事件、属性、方法等关键性概念作详细的讨论,以使读者能准确理解和掌握 VB 的面向对象程序设计思想。

1.3.1 对象

所谓“对象”,其实就是日常生活中的一个普通术语,现实生活中每个可见的实体,如每个人、每本书、每张桌子等,均可视为一个对象,整个世界就是由各式各样不断运动着的对象组成的。

用 VB 开发应用程序,首先就要学会从“对象”的角度来看整个程序设计过程。在 VB 中,可视化对象分为两大类,即窗体和控件。窗体就是窗口本身,控件则是在窗体上构成用户图形界面的一些基本组成部件,如命令按钮、文本框等。VB 应用面向对象的程序设计方法,把程序和数据封装起来作为一个对象,并为每个对象赋予应有的属性,使对象成为实在的东西。在设计对象时,单击工具箱中工具按钮即可在界面上画出所需的控件对象,此时,VB 自动生成建立对象的程序代码、初始化该对象的属性并将它们封装起来。每个对象以可视化图形方式显示在界面上。如图 1-1 所示的整个画面就是由一个窗体对象和若干个控件对象组成的可视化对象。

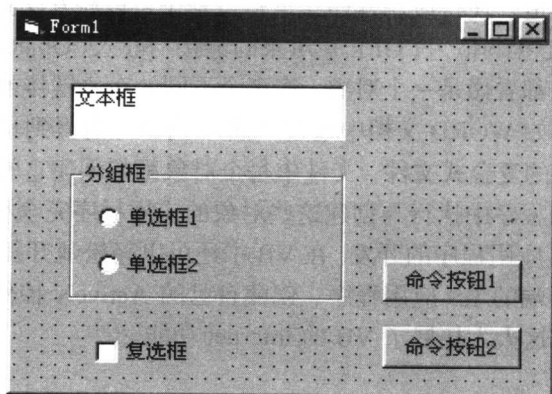


图 1-1 窗体对象及其上的控件对象

1.3.2 属性

属性用于描述对象的状态或特征。例如，对于一个人，可以使用姓名、性别、年龄、职业等特征来描述；对于一台电脑，可以用 CPU 的型号、内存容量、硬盘容量、显示器尺寸等特征来描述，所有这些特征称其为属性。在 VB 中，窗体和控件等对象也具有各自的属性。以窗体对象为例，它具有标题（Caption）、名称（Name）、背景颜色（BackColor）、前景颜色（ForeColor）、字体（Font）、是否可见（Visible）等属性。

由于事物的多面性，一个对象的属性也往往有多个，不同的对象有着不同的属性集合，但也有一些属性是每个对象都有的（如 Name 属性）。一个对象的所有属性集合称之为属性表，在 VB 集成开发环境中，当选中一个对象后，该对象的所有属性及属性值就会在“属性窗口”中显示出来，以方便用户查看或修改。图 1-2 反映了一个窗体对象的所有属性，同时，在属性窗口的下部有属性的对应描述。

窗体或控件对象刚被创建时，系统会自动赋予该对象默认的属性值。在对窗体或控件进行编程之前，常需对其部分属性重新设置。在 VB 中可以通过两种方式来设置对象的属性，一种是交互方式，另一种是编程方式。交互方式是在设计时通过属性窗口进行属性设置，这在第二章作详细介绍。这里，首先介绍编程方式，它是用程序代码设置对象属性。



图 1-2 窗体对象的属性

设置属性的语句格式如下：

[对象名.]属性名 = 属性值

其中，对象名即对象的名称，在 VB 中，对象的名称由其 Name 属性决定。

例如，假定一个窗体的名称为 Form1，它的 Caption 属性表明其标题栏上显示的内容，则语句：

Form1.Caption="我的窗体"

将使窗体 Form1 的标题栏上显示“我的窗体”。

设置属性时一定要明确操作的是哪一个对象，不要写错对象名。省略对象名时指当前窗

体。例如：

```
Caption = "我的窗体"
```

表示在当前窗体的标题栏显示字符串“我的窗体”。

1.3.3 事件

在 VB 中, 每个对象都能接收多个不同的事件, 并能通过程序代码对这些事件做出响应。在没有事件发生时, 整个程序处于停滞待命状态, 只有当事件发生时, 程序才会继续运行。

所谓事件, 是指由 VB 预先设置的、能够被对象识别的动作。例如, 鼠标在窗体上单击一下, 就会在该窗体上产生一次单击事件; 若鼠标在窗体上移动时, 则会在该窗体产生一个鼠标移动事件等。在 VB 中, 控件对象常用事件有 Click (单击)、DblClick (双击)、KeyPress (键盘按键)、MouseMove (鼠标移动)、LostFocus (失去焦点)、Change (改变) 等。

当事件由用户触发 (如 Click) 或由系统触发 (如 Load) 时, 对象就会对该事件做出响应。响应某个事件后所执行的操作是通过一段程序代码实现的, 这段程序代码称为“事件过程”。如果希望某个对象在接收某个事件之后能作出预期的反应, 在该对象的特定事件过程中则应编写相对应的程序。

事件过程的一般格式为:

```
Sub 对象名__事件名 ( )
```

```
    (事件过程的内容)
```

```
End Sub
```

其中, Sub 为定义过程开始的语句, End Sub 为定义过程结束的语句。

例如, 如果需要单击一个命令按钮后, 在窗体 Form1 上显示一个“Hello, 您好!”字符串, 则可以编写如下事件驱动程序:

```
Sub Command1_Click()
```

```
    Form1.Print "Hello, 您好! "
```

```
End Sub
```

在上述事件中, 所有代码就是一个事件过程。该事件过程赋予了一个对象 (命令按钮) 在该事件发生时的具体行为。

1.3.4 方法

在面向对象程序设计 (OOP) 中, 引入了称为方法 (Method) 的特殊过程或函数。方法是指各种可在对象上进行的操作, 它是对象本身所内含的函数或过程 (注意: 不是事件过程)。当用户用“方法”来控制某个对象时, 其实就是执行该对象内部的某个函数或过程。

方法的操作类似于过程和函数, 其调用格式为:

```
[对象名.]方法名 [参数表]
```

其中的“参数表”不是每个方法都具有的。有些方法必须带参数, 如 Zorder、Move 等; 有些方法参数可选, 如 Print、Show 等; 而有的方法则不需要参数, 如 Cls、Hide 等。

例如, 上例中的语句 Form1.Print "Hello, 您好!" 就使用了窗体对象 Form1 的 Print 方