

● 计算机及软件技术丛书 ●

学习和使用Visual BASIC

韩 羽 何志鹏 李 莉 等编

潘金贵 顾铁成 审校



南京大学出版社

《计算机及软件技术丛书》编委会

学术顾问 孙钟秀 张福炎 郑国梁

主 编 谢 立

副主编 时惠荣 潘金贵 丁 益 赵沁平

编 委 (按姓氏笔画为序)

丁 益	丁嘉种	王永成	孙志挥
时惠荣	陈 禹	陈道蓄	赵沁平
杨静宇	钱士钧	钱培德	徐宝文
顾其兵	谢 立	潘金贵	

出版者的话

我国社会主义经济建设的蓬勃发展,极大地推动着社会信息化的进程,也促进了我国信息产业的发展。现在,计算机的应用已渗透到社会和生活的各个领域。作为社会信息化基础的计算机及软件技术,正为越来越多的人掌握和应用,计算机及软件技术也因此而不断更新、发展。

掌握计算机技术,是现代入特别是跨世纪的中青年人在当今激烈的社会竞争中制胜的基础,也是未来信息化社会对每个人的要求。然而,在我国,计算机基础教育尚欠普及,计算机特别是微型计算机及软件技术的应用和开发也还处在一个较低的层次。许多非专业人员希望能使用计算机,但面对纷繁的专业知识,众多的技术资料,视学习计算机的使用为畏途,专业人员面对软件技术的快速更新,目不暇接。为了让更多的人熟悉计算机技术,利用计算机服务于自己的管理、科研、教学工作,使我国的计算机及软件技术的应用和开发紧随国际潮流,普及和提高我国计算机应用和开发的水平,我们为此组织编写并陆续出版《计算机及软件技术丛书》。

本《丛书》将以应用为基础,兼顾普及与提高。组织科研、教学和应用开发第一线的专家、学者,结合国外计算机及软件技术的最新发展和趋向与国内的应用现状和方向,为初学者提供系统的入门读物,为专业人员介绍适合国情的最新实用技术,既有理论性、学术性强的专著、专论,也有普及性、实用性的教材、手册,以满足多层次读者的需要。

本《丛书》的编写将立足于现实,着眼于未来,力争反映国内外计算机及软件技术的最新动态和发展趋向,引导和帮助读者学习、吸收、掌握计算机的新理论、新技术和新成果。

我们将根据读者需要,不断充实、完善本《丛书》内容,同时诚恳欢迎读者对本《丛书》提出建议、批评,也热忱欢迎向本《丛书》赐稿。

南京大学出版社

《计算机及软件技术丛书》编委会

前 言

这是一本非常实用的适合 3.0 以上版本的 Visual Basic 应用开发指南。内容覆盖控件设计、用户化窗体、用户化界面、基本编程语法、流程控制、过程、数组、记录的设计、界面设计、程序测试、绘图技巧、鼠标设计,以及动态链接库的设计和技巧。本书内容由浅入深,可读性强,适用于初中级水平的读者,专业读者也可将本书作为参考手册。

Basic 语言是一种适用性强、普及面广的程序设计语言,它的语法核心很小,外延性大,因此几乎所有编译系统都对它提供支持。小至计算器、单片机,大到大型计算机,都用到 Basic。作为全世界最大的微机软件制造商的美国微软公司,更是把 Basic 语言作为重要的程序设计语言加以实现。Visual Basic 是微软公司推出的面向 Windows 程序设计的可视化 Basic 程序设计集成环境,它将较为复杂的 Windows 程序设计简化成初学者和非计算机专业用户容易接受的 Basic 程序设计,这无疑给广大用户提供了在 Windows 环境下开发各自领域的应用程序的能力。

本书的特点在于引导初学者进入 Visual Basic 程序设计的美妙境界,它抓住了初学程序设计者必须掌握的几个关键内容:语言集成环境、语法、函数及过程设计、简单的数据结构及调试方法,同时通过图形设计和动态链接库设计等实用有趣的技术,使读者的综合技能得到加强。

本书主要由韩羽编写,参加编写的还有何志强、李莉、胡志坚、西门旭、顾仁、方宪子等。何胜利对全书进行了初校,最后由潘金贵和顾铁成审校定稿。限于水平,书中若有不足之处,望读者批评指正。

编 者

目 录

第一章 Visual Basic 入门	1
1.1 安装和启动 Visual Basic	1
1.2 Visual Basic 的屏幕布局	1
1.2.1 主窗口	3
1.2.2 工程窗口	3
1.2.3 窗体窗口和工具箱	4
1.2.4 属性窗口	5
1.3 创建用户界面	5
1.3.1 设置窗体的属性	6
1.3.2 增加显示对象	7
1.3.3 完成窗体设计	8
1.4 编写代码	10
1.5 运行程序	14
1.6 设计改进	14
1.7 保存程序	16
1.8 编译 Stopwatch 程序	17
第二章 控件设计	19
2.1 定制复选框	19
2.2 实时添加控件	21
2.3 设计含有驱动器、目录和文件的列表框	25
2.4 将 Windows 消息快速传给控件	28
2.5 清除列表框	29
2.6 利用 API 设计文件对话框	31
2.6.1 捕获错误	37
2.6.2 Form_Load 程序	37
2.6.3 文件对话框的测试	38
2.7 利用 API 设计文本编辑程序	38
2.8 设计传统的文件对话框	45
2.9 设计可多重输入的任务表	50
2.10 设计带简单查找功能的列表框	52
2.11 利用 API 设计带查找功能的列表框	55
2.12 设计带运行连续事件功能的按钮	57
第三章 定制窗体	60
3.1 一个新的工程文件实例	60
3.2 窗体的属性	61

3.2.1 Properties 窗口	62
3.2.2 窗体的最一般属性	65
3.2.3 尺寸属性	67
3.2.4 Icons(图标)	68
3.3 颜色属性	69
3.3.1 通过调色板访问 BackColor 及 ForeColor	70
3.3.2 十六进制表示的颜色	72
3.4 让窗体响应事件	73
3.5 打印窗体	79
3.6 保存已做的工作	81
3.6.1 在 File 菜单中进行保存	81
3.6.2 只保存代码	83
3.7 建立独立的 Windows 程序	83
第四章 建立用户界面	86
4.1 工具箱	86
4.2 创建控件	88
4.2.1 操作已存在的控件	89
4.2.2 生成控件的捷径	90
4.2.3 操作多个控件	90
4.2.4 删除控件	91
4.3 命令按钮的属性	91
4.3.1 标题属性(Caption)	92
4.3.2 控件名字属性	93
4.3.3 命令按钮的其它有用属性	94
4.3.4 用于设置属性的捷径	95
4.4 为命令按钮编写一些简单的事件驱动程序	95
4.4.1 命令按钮的其它事件	98
4.4.2 关于命令按钮的其它几个问题	99
4.5 加速键	99
4.6 图像控件	99
4.7 线段及图形控件	102
4.7.1 图形控件	102
4.7.2 线段控件	104
4.8 文本框和标签	104
4.8.1 文本框的标准属性	104
4.8.2 文本框的一些特殊属性	104
4.8.3 文本框的事件驱动程序	106
4.8.4 标签(Labels)	106
4.8.5 标签的几个有用属性	106
4.8.6 标签的事件驱动程序	107
4.9 如何选中需要的控件	108

4.10	消息框	108
4.11	栅格	110
4.12	Visual Basic 应用程序运行时发生的情况	111
4.13	窗体的 AutoRedraw 属性及 Refresh 方法	113
4.14	窗体的 ASCII 码方式表述	114
第五章	编程入门	116
5.1	Visual Basic 程序解析	116
5.2	代码窗口	116
5.3	Visual Basic 中的语句	117
5.3.1	注释语句	117
5.3.2	End 语句	118
5.4	赋值语句和属性设置	118
5.5	变量	120
5.5.1	变量类型	121
5.5.2	变量详述	122
5.5.3	类型的 Dim 语句	123
5.5.4	改变类型的缺省值	124
5.5.5	变量需要的声明	124
5.5.6	变量的作用域	125
5.6	过程间值的共享	125
5.7	字符串	127
5.7.1	ASCII/ANSI 代码	127
5.7.2	固定长度的字符串	129
5.8	数值	129
5.8.1	有关数值的操作	130
5.8.2	关于数值的进一步说明	132
5.9	一个计算程序实例	133
5.10	常量及常量文件	136
5.11	具有多窗体的工程文件	138
5.11.1	为多窗体编写代码	138
5.11.2	如何在运行时刻处理多个窗体	140
5.11.3	多窗体使用实例	141
5.11.4	在窗体(模式)中保存焦点	142
5.11.5	输入框	142
5.12	在窗体中打印	144
5.12.1	在窗体中居中文本的程序举例	144
5.12.2	代码中的字体属性	145
5.12.3	表	146
5.13	打印机及其有用属性和方法	147
5.14	Format \$ 命令	148
5.15	日历信息	150

5.15.1	Date\$ 函数	150
5.15.2	数值型日历函数	150
5.15.3	日期函数举例	151
5.15.4	其它日历函数	152
第六章	控制程序流程	154
6.1	重复操作	154
6.1.1	确定循环	154
6.1.2	非确定循环	160
6.2	选择语句	166
6.2.1	If-Then 块	168
6.2.2	If-Then 简介	169
6.2.3	Dir\$ 命令实例	170
6.2.4	组合 If-Then 和循环	170
6.2.5	Select Case 语句	171
6.2.6	结束 If-Then	172
6.2.7	按键过程实例	174
第七章	内部函数	176
7.1	字符串函数	176
7.1.1	字符串分析函数	176
7.1.2	Rnd 函数	185
7.2	位处理及位级逻辑操作符	191
7.3	金融函数	196
7.4	数值函数	199
第八章	过程和错误捕获	202
8.1	函数过程	202
8.2	函数的高级用法	205
8.2.1	提前退出函数	206
8.2.2	字符串函数的例子	207
8.2.3	文本分析	209
8.2.4	PigLatin 转换器实例	210
8.3	子过程	212
8.4	过程和函数的高级用法及实例	217
8.5	传递控件和窗体信息	222
8.6	访问 Windows 函数	225
8.7	代码模块:全局过程和全局变量	227
8.7.1	添加或删除代码模块	228
8.7.2	过程范围的进一步讨论	229
8.8	DoEvents 函数	230
8.9	错误捕获机制	231
8.10	有关程序设计方面的一些建议	233
第九章	数组、栅格和记录	236

9.1 控件数组	236
9.1.1 增加和删除控件数组中的控件	238
9.1.2 标号的一个方阵控件数组实例	240
9.2 一维数组列表	243
9.2.1 固定和动态列表	243
9.2.2 使用列表的一些方法	245
9.2.3 一个改善嵌入式的随机数字发生器的实例	246
9.2.4 带有索引范围的列表	247
9.2.5 Erase 语句	248
9.3 多维数组及魔方实例	248
9.4 过程中使用列表和数组	250
9.5 栅格控件	253
9.5.1 栅格控件的一般属性	254
9.5.2 栅格控件的事件和方法	256
9.6 排序和查找	257
9.6.1 查找	257
9.6.2 排序	259
9.7 记录	264
第十章 完成用户界面	267
10.1 工具箱的进一步讨论	267
10.1.1 框架控件	269
10.1.2 生成控件的快速方法	270
10.1.3 图片框	270
10.1.4 图形与图片框	271
10.1.5 向工程中加入工具条和状态条	272
10.1.6 单选按钮	274
10.1.7 复选框	275
10.1.8 列表框和组合框	276
10.1.9 滚动条	282
10.1.10 定时器	285
10.2 菜单	288
10.2.1 菜单设计窗口应用举例	292
10.2.2 弹出式菜单	292
10.2.3 在运行时操作菜单的实例	293
10.3 常用对话框及其使用	295
10.4 MDI 窗体	300
10.5 如何显示控件的进一步讨论	301
10.6 关于窗口设计的一些建议	301
第十一章 测试和调试的工具及技术	303
11.1 调试工具及其功能	303
11.2 测试程序	304

11.3	使测试简单的程序设计.....	305
11.4	Debug 窗口	306
11.4.1	在 Debug 窗口内进一步调试程序	307
11.4.2	单步执行	308
11.5	暂停程序.....	309
11.6	程序调试综述.....	314
11.7	文档和程序风格.....	315
第十二章	绘图操作简介.....	317
12.1	绘图基础.....	319
12.2	窗体标度的改变与定制.....	320
12.3	色彩.....	324
12.4	像素控制.....	325
12.5	线和框.....	330
12.5.1	X-Y 平面实例	332
12.5.2	栅格图	333
12.5.3	DrawWidth 和 DrawStyle 属性	334
12.5.4	方框	336
12.5.5	修饰控件实例	340
12.6	圆、椭圆和饼图	342
12.6.1	饼图实例	345
12.6.2	椭圆和解析度	345
12.7	曲线.....	346
12.7.1	简单公式描述的图形	346
12.7.2	极坐标	350
12.7.3	极坐标演示例程	353
第十三章	监视鼠标活动.....	358
13.1	鼠标事件过程.....	358
13.1.1	MouseUp/MouseDown 事件	359
13.1.2	MouseMove 事件	361
13.2	拖动和放置操作.....	363
13.2.1	手工拖动	364
13.2.2	DragOver 事件	365
13.2.3	以拖/放方式删除文件实例	366

第一章 Visual Basic 入门

为了说明 Visual Basic 的易学性及其编写应用程序的方便性,本章将直接开始讨论编程,并介绍一个简单的程序。从使用中学习 Visual Basic,体验 Visual Basic 诱人的魅力。

1.1 安装和启动 Visual Basic

Visual Basic 有一个名为 `SETUP.EXE` 的程序,它负责将 Visual Basic 安装在计算机的硬盘上,Visual Basic 的文档详细描述了整个安装过程,但实际上只执行以下四个步骤:

(1) 在 MS-DOS 提示下键入 `Win` 并回车,启动 Windows。Visual Basic 可在 Microsoft Windows 3.0 以上的版本中运行。

(2) 在软盘驱动器中插入标有“Disk 1”(1 号盘)的软盘。

(3) 从 Windows 程序管理器(Program Manager)的 File(文件)菜单中选择 Run(运行)命令。如果 Disk 1 插在 A 驱动器中,则键入 `a:setup` 和回车键;如果使用驱动器 B,则键入 `b:setup` 和回车键。

(4) 根据屏幕上的指令,回答 Setup 提出的各种安装选项问题。

在回答完所有问题后,Setup 程序将把 Visual Basic 安装在计算机的硬盘上,并在程序管理器窗口中加入一个 Visual Basic 组窗口的图标。

启动 Visual Basic 最简单的方法就是在程序管理器窗口中双击 Visual Basic 图标,也可以双击 Windows 文件管理器窗口中代表 Visual Basic 执行文件 `VB.EXE` 的图标。另外,还可以通过在 MS-DOS 提示下键入 `win vb` 命令,同时启动 Windows 和 Visual Basic。

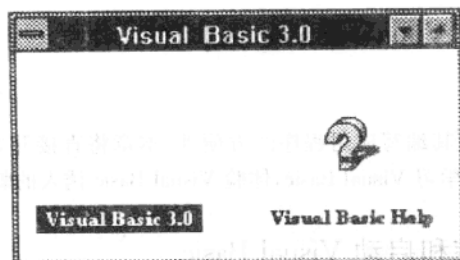
1.2 Visual Basic 的屏幕布局

上一节所述的启动 Visual Basic 的三种方法可参见图 1.1。

启动 Visual Basic 之后,屏幕上会出现五个窗口,如图 1.2 所示(在用户的 Visual Basic 屏幕中,有些窗口可能会重叠。为了清楚地观察各个窗口,图 1.2 中的窗口已进行缩放和重新定位)。屏幕的顶部是主窗口,它包含标准的文件(File)和编辑(Edit)菜单以及 Visual Basic 的其它菜单与工具条。屏幕中心是窗体(form)窗口,即标题为 `Form1` 的空字符串,在其左边出现的是一个称为工具箱的调色板式的窗口,窗体窗口的右边是属性(Properties)窗口,而属性窗口之下是工程(Project)窗口。

注意,在图 1.2 中,除 Visual Basic 之外的其它应用程序都缩成了图标。将应用程序缩小成图标可以解决屏幕混乱的问题。当从程序管理器的选项(Options)菜单中选择 `Minimize On Use` 选项后启动新程序时,程序管理器窗口自动缩小成图标。

下面让我们分别来讨论这五个窗口。



第一章

C:\>win.vb

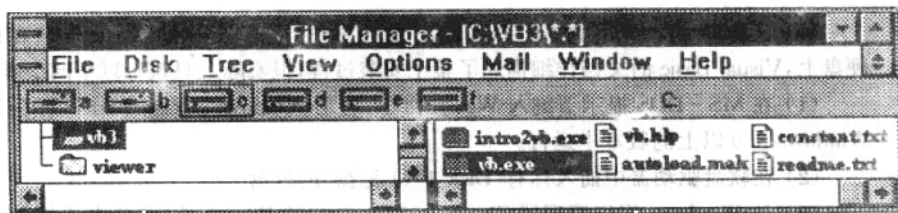


图 1.1 启动 Visual Basic 的三种方法

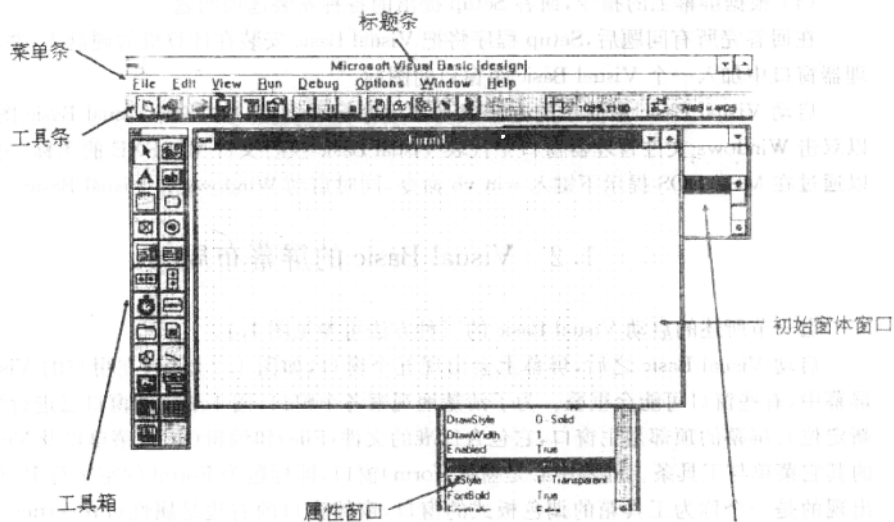


图 1.2 Visual Basic 屏幕

1.2.1 主窗口

主窗口包含菜单条,它有八个下拉式菜单。其中最重要的(尤其是学习 Visual Basic 的时候)是帮助(Help)菜单,从该菜单中可以查阅 Visual Basic 的使用指导,找到与 Microsoft 产品支持服务机构的联系方法。下面讨论复杂的 Visual Basic 联机帮助系统。

当选择 Help 菜单中的 Contents 时,我们将看到联机帮助系统中所含信息的分类表。用鼠标单击其中的下划线词组,可以移动到需要查阅的特定段落和主题。当选择 Help 菜单中的 Search(搜索)命令时,会显示一个对话框,让用户指定一个特定的主题,并马上显示该主题的帮助信息。

另外,Help 还提供了与上下文相关的帮助信息。若想了解关于按钮、对话框、窗口、错误信息或 Visual Basic 中的其它构件,只要在该构件加亮显示时按 F1 键,帮助系统将立即显示相关的帮助信息。

主窗口还包含工具条,它是 Visual Basic 3.0 版才加入的新功能。图 1.3 中工具条上的按钮是常用命令的简捷按钮。例如,不必打开 File 菜单再选择 Open Project 命令,只需单击 Open Project 按钮即可。

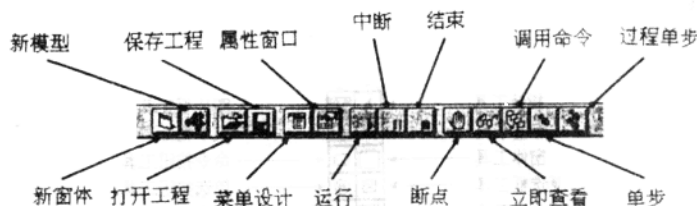


图 1.3 Visual Basic 工具条

最后,主窗口工具条右边有两个显示域,它们分别指标了窗体窗口中当前被选对象的位置和大小。

1.2.2 工程窗口

工程窗口包含了运行正在编写的 Visual Basic 程序所需的文件清单。虽然我们还没有开始设计程序,工程窗口中已出现了三个文件名,如图 1.2 所示。第一个文件名是 Form1.frm,右侧的标签(Form1)表明了该文件是与称为 Form1 的窗体相关联的。如果不改变名字就在磁盘上存储窗体,Visual Basic 将使用缺省文件名 Form1.frm。

一个应用程序可由多个窗体组成,每一个窗体都可存储在一个文件中(因为文件是独立的,所以,其中的窗体也可以由其它应用程序共享)。

工程窗口中的其它两个项是文件名 GRID.VBX 和 OLECLIEN.VBX。文件扩展名 VBX 表示这些文件是 Visual Basic 的扩展文件。当装载一扩展文件时,工具箱中会增加一些辅助工具(在 1.2.3 节中讨论)。

工程窗口还包含两个按钮,标签名分别为 View Form(查看窗体)和 View Code(查看代

码)。在工程窗口中选择一个文件时, Visual Basic 将显示相应的窗体, 窗体可用于设计应用程序的用户界面, 即应用程序中用户可以看见并与之交互的部分。如果按下工程窗口中的 View Code 按钮, 选中文件的代码将出现在不同的窗口中。代码指的是编程语言中的语句, 因此, 编程的过程也称为编码。用 Visual Basic 创建程序时, 工作可以分成两部分, 即窗体设计(用户界面)和编码(控制程序的操作)。如果要从代码窗口切换到窗体窗口, 只要在窗体上单击一下, 也可按工程窗口中的 View Form 按钮。

1.2.3 窗体窗口和工具箱

窗体是一个对应于应用程序运行时所见的窗口的显示区域。当启动一个新工程时, Visual Basic 创建一个空的窗体, 并命名为 Form1。在设计应用程序时, 窗体就像一块画布, 在画布上可以画出组成应用程序的各个构件, 窗体上的应用程序构件称为对象(object)或控件(control), 如图片框、单选按钮和滚动条; 另外, Visual Basic 将窗体本身也视为对象。

控件创建于工具箱调色板, 如图 1.4 所示。每个控件由一个工具图标(简称为工具)表示。图 1.4 中的大多数工具都是 Visual Basic 固有的。当然, 工具箱可以扩充, 以包含新的工具。工程窗口中扩展名为 VBX 的文件均为工具箱提供了一种或多种新工具。例如, 工具 Grid 和 OLE Client 分别由文件 GRID.VBX 和 OLECLN.VBX 提供, 它们是工程文件指定的缺省工具。

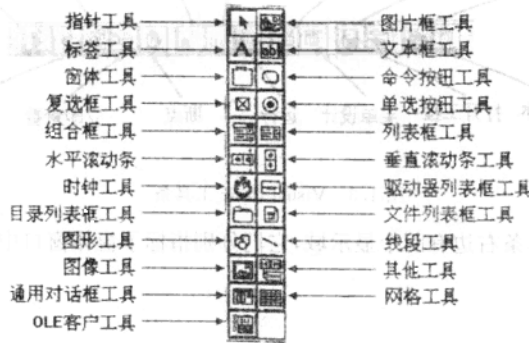


图 1.4 Visual Basic 工具箱

Microsoft Visual Basic Professional Edition(专业版)提供了更多的 VBX 文件, 并允许程序员设计新的工具(对于 C 或 Pascal 程序员)。同时, 第三方软件开发商还提供了大量其它工具。

通过从工具箱中选择控件并把它们放到窗体中, 即可设计程序的外貌。在设计应用程序时, Visual Basic 的操作和运行该应用程序时是不一样的。在设计阶段, Visual Basic 提供了用于创建显示对象和编写程序的工具。在窗体上改变对象的大小、位置和其它属性, 可以控制对象的外观和行为, 但这些对象不是活动的, 代码也不执行。例如, 在窗体上放置一个滚动条(scroll bar)对象后, 我们可以改变该对象的大小和位置, 但不能用它滚动任何东西。

当然,在运行编写完成后的应用程序时,Visual Basic 会移走设计工具。窗口布局是设计阶段的窗体,屏幕上的对象可以被激活,用户可以按我们设计的方式与显示区进行交互。应用程序一旦启动,Visual Basic 就开始执行程序语句(在主窗口的标题条中,Visual Basic 会询问是要设计应用程序还是运行应用程序)。

1.2.4 属性窗口

Visual Basic 的属性(Properties)是描述对象性质的形式化机制。在现实生活中,我们可问“这件上衣是什么颜色”,而在 Visual Basic 中,问题则变为“这件上衣的颜色属性值是多少?”,这两个问题的答案当然都一样,即是“红色”。品种是上衣的另一种典型属性,品种属性的期望值(或设置值)可能包括西装、休闲装和茄克等。

每个 Visual Basic 对象都有特殊的属性,其设置值控制了该对象的外观和行为。有些属性仅限于一定的值,例如,对象的可见性只能设置为 True 或 False(即可见或不可见)。其它的属性(如某窗体窗口听标题)几乎可设为任意正文。注意,不必设置每个对象的所有属性,许多属性可采用其缺省值。

虽然在设计阶段和应用程序运行时可改变许多属性,但属性窗口(如图 1.5 所示)只有在设计阶段才是活动的。激活属性窗口有多种方法,包括简单地单击属性窗口,在 Windows 菜单中选择 Properties 命令,按 F4,或选择工具条上的 Properties Windows 按钮。

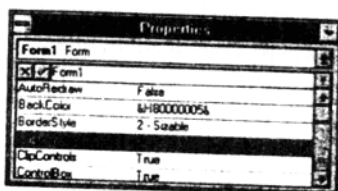


图 1.5 属性窗口

属性窗口顶部的下拉式列表框称为对象框,它显示了应用程序中每个对象的名字及对象的类型。初始时,对象框只包含该窗体的信息,但将控件加入到窗体中时,Visual Basic 会把这些对象相应地加入到对象框的下拉列表中。

位于对象框之下的是设置框和属性列表,属性列表允许滚动对象框中显示的对象的属性,以但观察每项属性的当前设置值。当从该列表中选择一项属性时,其当前设置值会出现在属性列表上的设置框中。如果希望改变设置值,可以根据特定的属性,在设置框中键入一个新的条目,或从下拉列表中选择一个新的、预先定义的设置值。

下面我们举例说明怎样用各种 Visual Basic 窗口来创建应用程序。

1.3 创建用户界面

作为第一个例子,让我们创建一个简单的程序,它只记录走过的时间,就像一只秒表一样。秒表有一个计时启动按钮,还有一个计时停止按钮,在表的正面可以观察走过的时间。这

种模型可以作为程序的基础,在 Visual Basic 中,窗体相当于秒表表面,在窗体上面也有启动按钮和停止按钮,在开始设计应用程序之前,首先需要检查和修改一些窗体的属性值。

1.3.1 设置窗体的属性

在运行应用程序时,窗体将作为一个标准应用程序窗口出现。如果希望该窗口类似于其它 Windows 应用程序显示的窗口,则应该使它具有类似的属性,例如,Windows 应用程序一个共同的属性就是应用程序的名字作为其标题出现在其标题条中。在 Visual Basic 中,许多这样的属性均由对象的属性来控制。

在开始设置窗体的属性时,首先要在标题名为 Form1 窗口中的任何地方按一下鼠标键,选择该窗体,使其成为当前对象。而在 Visual Basic 的属性窗口,Form1 被显示在对象框中。如果要改变窗体的标题,应单击属性列表中的标题(Caption)属性。当前(缺省)的标题 Form1 将出现在设置框中,键入 Stopwatch,将它作为新的标题,在键入它后,这新标题会显示在窗体窗口的设置框和标题条中。

注意设置框中左边的两个按钮,它们分别带一个“×”和“√”标签。“√”按钮用于确认设置框中所输入的设置值,按下“√”按钮等效于按回车(Enter)键。按“×”按钮则取消当前输入,并在设置框中恢复以前的设置值。

当选中的属性只能接受一组有限的预定义值时,设置框的作用就像一个下拉列表一样,而不是作为一个正文输入域。例如,滚动属性列表,并选择可见(Visible)属性,现在,设置框边上的箭头按钮已激活,说明可见属性的设置值仅限于下拉列表中的值。如果单击向下箭头按钮,该列表将拉下来。以显示设置值 True 和 False,如图 1.6 所示。注意,可见时,属性应设为 True。

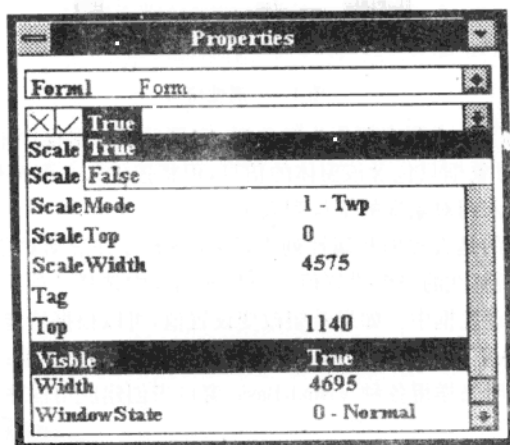


图 1.6 设置窗体的可见属性

和窗体外貌有关的另一属性是 BorderStyle(边框类型),它控制着运行应用程序时用户是否可以改变显示窗口的大小。因为秒表应用程序将有一个固定的外貌,其窗口的大小不允

许改变,所以,应从属性列表中选择 BorderStyle 属性,并从设置框的下拉列表中选择 3—Fixed Double,将该属性设为 3—Fixed Double 值(Fixed Double 表示窗口的边框大小不能改变,并且不带最小和最大按钮)。

Name(名称)属性

每个 Visual Basic 对象都有一个称为 Name 的属性。当设置 Name(名称)属性时,即给对象取一个标识名,在程序中可用它表示对象。在窗体窗口中,单击一个对象就可以访问该对象;但在程序的代码中,访问对象必须使用名称属性中所指定的名字(注意,Name 属性和 Caption 属性是不同的。对象名称是程序代码中的标识符,而对象标题是用户在屏幕上所见到的——即在应用程序窗口中显示的标识正文)。在属性列表中,选择窗体的名称属性,其缺省值为 Form1,它显示在设置框中。用户在设置框中键入正文 MyForm,以改变窗体的 Name 属性。

注意,当改变窗体的名称时,工程窗口中显示的文件名列表中会反映这一变化。请记住,每个窗体的定义存储在一个独立的文件中。当保存应用程序时,用户可以根据个人的喜好来选择文件名,但 Visual Basic 会根据窗体名称,为用户推荐一个缺省文件名。

1.3.2 增加显示对象

现在让我们返回窗体的设计,创建用于启动和停止计时器(秒表)的两个按钮,步骤如下:

(1) 单击工具箱中的命令按钮(Command Button)工具,再将鼠标移到空的窗体窗口。这时,光标会变成十字光标,表示正处于绘图模式:

(2) 将光标放在窗体的左上角;

(3) 按住鼠标器按钮不放,将光标向右下角拖动;

(4) 放开鼠标器按钮。

在放开鼠标器按钮之后,Visual Basic 在拖动操作所定义的矩形区域内创建一个命令按钮对象,如图 1.7 所示。

在创建命令按钮后,工具箱中的箭头工具又变为了活动的工具。当箭头工具激活时,我们可以移动对象和改变对象的大小。若要移动对象,只需把它拖动到新位置;如果要改变对象的大小,可拖动它的柄(如图 1.7 所示)以扩大或缩小对象;如果柄不可见,可单击该对象,这时它会出现柄。

现在,开始创建第二个命令按钮,这次使用一种更简单的方法(简捷方法)。先双击工具箱中的命令按钮工具,Visual Basic 会创建一个缺省大小的按钮,并把它放在窗体的中心位置。再移动并缩放第二个按钮,使两个按钮都在屏幕的左边,并且大小相同。

这时,两个按钮的缺省标签为 Command1 和 Command2,它们分别作为秒表应用程序的启动按钮和停止按钮。为了清楚起见,应赋予它们相应的标题。在设置这两个按钮的属性时,应遵循设置窗体属性同样的过程,即:

(1) 单击上方的命令按钮,选中按钮;

(2) 从属性窗口的属性列表中选择 Caption 属性;

(3) 在设置框中,将设置改变为 Start;