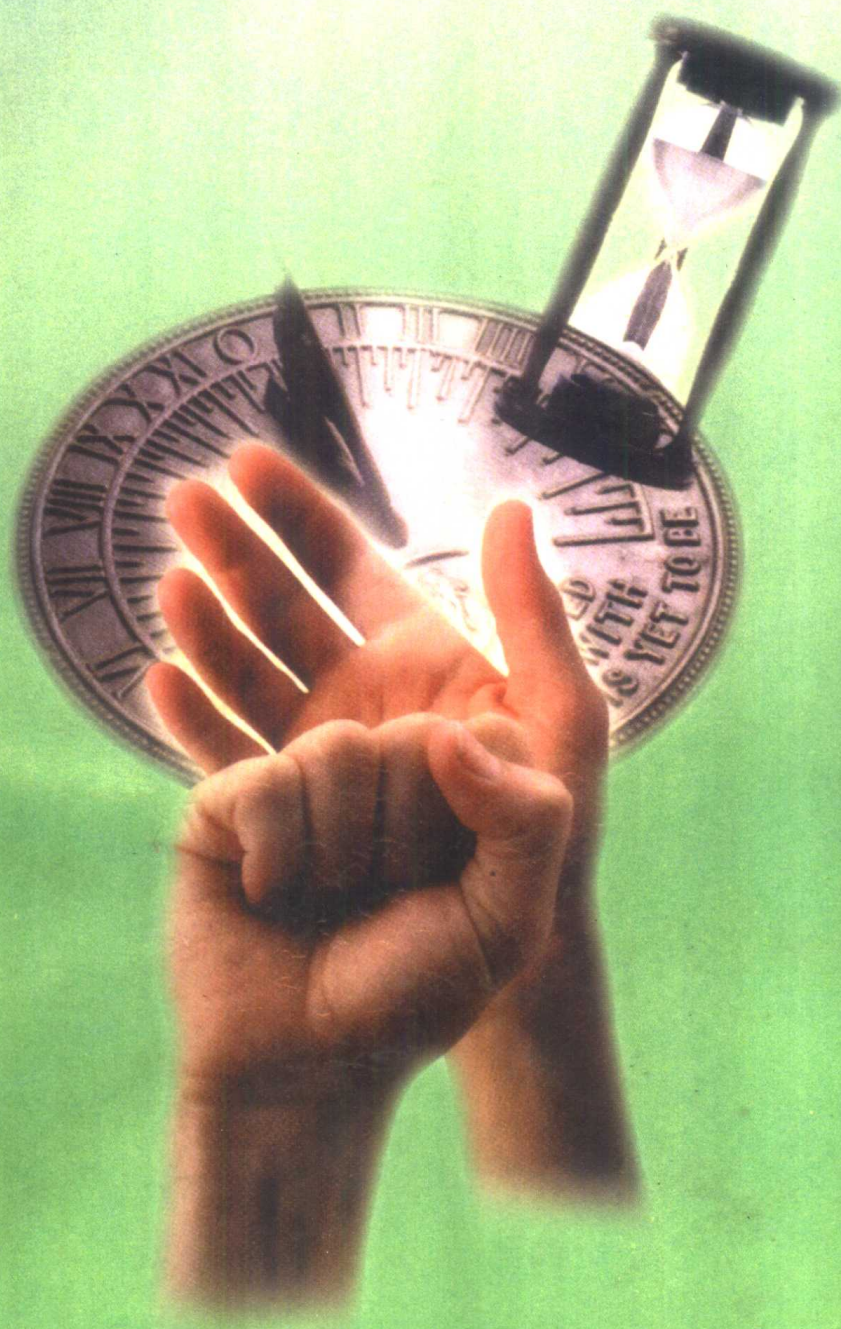


Visual Basic 4.0 for Windows

Visual Basic 4.0 for Windows

编程速成指南

赵越 朱尽染 等编著



同济大学出版社

Visual Basic 4.0 for Windows
编程速成指南

赵 越 朱尽染 等编
潘金贵 顾铁成 审校

同济大学出版社

内 容 提 要

本书从 Visual Basic 4.0 的基础内容讲起,逐一介绍了用户环境、窗体、用户界面、程序流程、函数、对象、测试、调试、文件操作、绘图,可以说几乎详细介绍了 Visual Basic 4.0 的所有方面。本书循序渐进,通俗易懂,相信本书能有助于读者很快地掌握 Visual Basic 4.0。

本书及其配套的《Visual Basic 4.0 for Windows 高级程序设计》是学习使用 Visual Basic 语言并希望快速掌握编程技术的用户的实用教材,也是 Windows 程序员和 Visual Basic 程序员必备的参考书。

责任编辑 王建中
封面设计 潘向葵

Visual Basic 4.0 for Windows 编程速成指南

赵 越 朱尽染 等编

潘金贵 顾铁成 审校

同济大学出版社出版

(上海四平路 1239 号,邮编:200092)

新华书店上海发行所发行

望亭电厂印刷厂印刷

开本:787×1092 1/16 印张:14.5 字数:370 千字

1997 年 10 月第 1 版 1997 年 10 月第 1 次印刷

印数:1-3000 定价:20.00 元

ISBN 7-5608-1875-7/TP·195

前 言

对于编写 Windows 应用程序的人来说,Microsoft 公司的 Visual Basic for Windows 编程环境的推出是一项令软件界欣慰的大事。Visual Basic 具有事件驱动的编程机制和新颖简明的可视化设计工具,并将 Windows 图形用户界面的特点发挥得淋漓尽致,为开发人员快速设计各种应用程序提供了有力的工具。

为了更好地理解 Visual Basic,我们不妨回顾一下 Visual Basic 的历史。

最早的编程语言产生于 1950 年,其目的主要是为了解决复杂的数学计算问题。对于普通人来说,几乎无法理解。但这在当时并没有什么问题,因为当时只需少数人会用这种语言,只有少数人才能使用计算机。但随着计算机软硬件技术的发展,计算机应用逐渐普及到国民经济各部门,甚至千家万户。这样,复杂的编程语言就成为计算机应用普及的一大障碍。有鉴于此,60 年代初,美国 Dartmouth 学院发明了一种名为 BASIC 的程序设计语言。最初的基本(Beginner's All-purpose Symbolic Instruction Code 的缩写)是一种非常简单的语言,其目的就是简化编程难度。大量的程序设计人员用 BASIC 语言编写了许多优秀的软件,包括阿波罗登月计划、矿产资源勘探软件等。

BASIC 的简单性也使得它很小,当微机的代表 Altair 于 1975 年在麻省理工学院诞生时,Microsoft 公司的创始人 Bill Gates 和 Paul Allen 承担了开发一个 BASIC 版本的任务,该 BASIC 版本后来发展成为个人微机中最流行的产品。

随着微机硬件水平的提高,BASIC 语言得到了进一步的发展,当最早的微机被 IBM PC 取代时,Microsoft 公司的 GW-BASIC 成为了标准。后来,对软件的快速化、小型化和易用化需求导致了 Microsoft Quick-BASIC 的诞生,Quick-BASIC 使 BASIC 语言成为 80 年代通用的几种高级编程语言之一,但随着微机图形功能和硬件功能的迅速增强,图形用户界面(GUI)诞生了,它是计算机史上的一次飞跃,使计算机技术跨入了一个崭新的时代。

Microsoft Windows 成为这一时代的典型代表,它为 PC 用户提供了一个直观的、界面友好的图形工作环境,用户只要简单地用鼠标按钮点一下菜单或图标,就可完成过去用键盘敲击几十次才能完成的工作,多任务、多窗口、即插即用等新技术使微机更易使用。尽管 Windows 环境对用户来说是很易于掌握的,但是对于程序员和希望自己设计一些专用应用程序的用户来说就困难多了。程序员必须编写程序来创建窗口、菜单、字体、对话框、消息框以及各种控件,同时还必须掌握各种资源,即使是编写简单的程序也需要大量参考资料,因此在程序员与用户之间形成了一种恶性分歧:一方面 Windows 程序的友好图形界面使应用程序越来越受用户欢迎,且用户对界面的要求越来越高;另一方面程序代码越来越大,程序员的编程难度越来越大,工作也越来越复杂,软件成本也不断上升。一个在屏幕上显示一条信息的简单程序在 DOS 下编写只需几行语句,而在 Windows 下编写类似程序需要 2-3 页代码;程序员还要学会如何控制字体、菜单、窗口、对话框、内存及其他系统资源。

一段时间以来,很多人不得不承认 Windows 的出现预示着业余程序员的末日。在 DOS 环境中,非计算机领域的专业人员一般都能学会编写简单应用程序的知识,如专业计算函数

15133/05

等,但在 Windows 中则望而却步,力不从心。

Visual Basic 1.0 版问世时,Microsoft 公司的总裁及首席执行官 Bill Gates 称它为“惊世骇俗的”。在《信息世界》(Infoworld)中,Steve Gibson 称 Visual Basic 为“令人震惊的新奇迹”,并且认为它会“极大地改变了人们对 Microsoft Windows 的看法以及使用 Windows 的方式”。Stewart Alsop 在接受《纽约时报》(New York Times)的采访时说它是“90 年代完美的编程环境”。

Visual Basic 的优点在哪呢? Visual Basic 到底是怎么一回事,它能为用户做些什么呢? 对此, Bill Gates 把它描述为“使用 Basic 语言的非常容易使用但功能强大的开发 Windows 应用程序的工具”。这句话的份量只有那些在过去为开发 Windows 应用程序饱尝痛苦的 C 程序员,才会真正体会到。

Visual Basic 2.0 版比 Visual Basic 1.0 版更快,更强大,而且更容易使用。Visual Basic 3.0 增加了一种对最强有力的可用数据资源的简单控制方法。Visual Basic 4.0 增加了对 Windows 95 和 Windows NT 的 32 位可执行程序开发的支持,并开始将 Visual Basic 转化为一种完全面向对象的程序设计语言。另外,与 Visual Basic 先前的版本不同,这种最新的版本允许用户方便地增加第三方的工具,使得开发过程更为容易。

本书介绍了目前最优秀的微机编程语言 Visual Basic 4.0。Visual Basic 4.0 的出现,使计算机编程步入了简单、快速的新纪元。本书抽取了 Visual Basic 4.0 的精华,并以简捷明了的方式介绍给读者。从入门、环境、表单设计到程序设计都一一作了介绍,并且书中探讨了用 Visual Basic 4.0 编程所必需的很多新的思维方法。只有充分了解了 Visual Basic 4.0 的一些核心细节,才能灵活运用 Visual Basic 4.0 的强大功能,所以无论是初学者还是传统编程语言方面的专家,读了本书都会获益匪浅。

参与本书编写的人员有:赵越、朱尽染、覃庭、程辉、刘石华、李梅、祖宁、王蕾、廖建、石海萍、刘诚、刘秀芳、党日生、梁金枝、陈志斌。全书先由吴佳教授和陈忠敏研究员作了初审,最后由潘金贵和顾铁成审校定稿。此外,朱闽虹为本书的编排付出了辛勤的劳动。在此对以上同志深表感谢。

限于水平和时间,书中的错误和不妥之处,敬请读者批评指正。

编者

目 录

第一章 概述	1
1.1 Visual Basic 对 Windows 的意义	1
1.1.1 基本配置	2
1.1.2 Visual Basic 4 的功能	2
1.2 基本操作	2
1.2.1 运行安装程序	3
1.2.2 运行 Visual Basic	3
1.2.3 如何使用 Visual Basic	4
1.2.4 运行 Visual Basic 项目	5
1.2.5 联机帮助系统	6
1.2.6 Visual Basic 应用程序开发概述	6
第二章 Visual Basic 4 用户环境	9
2.1 主屏幕及联机帮助系统	9
2.1.1 标题栏	10
2.1.2 菜单条	10
2.1.3 工具条概述	11
2.1.4 工具箱概述	11
2.1.5 初始的窗体窗口	13
2.1.6 Project 窗口概述	13
2.1.7 快捷(弹出式)菜单概述	14
2.1.8 联机帮助系统概述	14
2.2 File 菜单	18
2.3 Edit 菜单	20
2.4 View 菜单	21
2.5 Insert 菜单	22
2.6 Run 菜单	23
2.7 Tools 菜单	23
2.8 Add-Ins 菜单	25
第三章 控件、菜单和事件	27
3.1 如何操作 Properties 窗口	27
3.1.1 属性编辑器	27
3.1.2 下拉列表属性编辑器	27
3.1.3 对话框属性编辑器	28
3.2 窗体	29
3.2.1 窗体的属性	29
3.2.2 关于窗体的事件	32
3.3 控件处理技术	34
3.3.1 定制控件	34
3.3.2 标准控件的概览	35

3.3.3	专业版的控件	37
3.3.4	在窗体中添加控件	38
3.3.5	操作已存在的控件	39
3.3.6	焦点属性和事件	40
3.3.7	命令按钮、三维命令按钮和动画按钮	41
3.3.8	TextBox, RichTextBox 和 MaskedEdit 框	41
3.3.9	文本框的事件处理过程	43
3.3.10	标签	43
3.3.11	Listbox 和 ComboBox 控件	43
3.3.12	CheckBox 和 RadioButton 控件	44
3.3.13	计时器控件	44
3.3.14	Grid 控件	46
3.3.15	Grid 控件的事件	46
3.4	设计菜单	46
3.5	一个窗体的 ASCII 表示	49
第四章	设计用户界面	51
4.1	工具箱	51
4.2	生成控件	53
4.2.1	操作已存在的控件	53
4.2.2	生成控件的快捷键	55
4.2.3	多个控件的操作	55
4.2.4	加锁控件	55
4.2.5	删除控件	56
4.3	命令按钮的属性	56
4.3.1	Caption 属性	56
4.3.2	Name 属性	58
4.3.3	命令按钮的其他有用属性	58
4.3.4	设置属性的快捷键方法	59
4.4	为命令按钮编写简单的事件过程	59
4.4.1	命令按钮的其他事件	61
4.4.2	命令按钮的其他问题	61
4.5	快速访问键	62
4.6	图像控件	62
4.7	文本框	63
4.7.1	文本框的标准属性	64
4.7.2	文本框的几个特殊属性	64
4.7.3	文本框的事件过程	66
4.8	标签	66
4.8.1	标签几个有用的属性	67
4.8.2	标签的事件过程	67
4.9	浏览控件	68
4.10	网格	68
4.11	Visual Basic 应用程序的运行情况	69
4.11.1	Visual Basic 的显示	70

4.11.2	窗体的 AutoRedraw 属性	71
4.11.3	Refresh 方法	72
4.12	窗体的 ASCII 码表示方式	72
第五章	Visual Basic 程序设计基础	74
5.1	程序结构与 Code 窗口	74
5.1.1	Visual Basic 程序结构	74
5.1.2	Code 窗口介绍	74
5.2	Visual Basic 中的语句	75
5.2.1	注释语句	75
5.2.2	End 语句	76
5.3	常量、赋值和属性设置	76
5.3.1	常量	76
5.3.2	给属性赋值	77
5.4	变量	78
5.4.1	变量类型	79
5.4.2	关于变量的要注意的地方	80
5.4.3	关于类型的 Dim 语句	81
5.4.4	要求使用变量的声明	81
5.4.5	变量的作用域	82
5.4.6	在过程之间共享变量值	82
5.4.7	持久地拥有一个值	83
5.5	字符串	83
5.5.1	ASCII/ANSI 代码	84
5.5.2	固定长度字符串	85
5.6	数字	85
5.6.1	关于数字的操作	86
5.6.2	在 Visual Basic 中关于数字的进一步讨论	87
5.7	重复操作	88
5.7.1	定型循环	88
5.7.2	非定型循环	90
5.8	条件语句	92
5.8.1	它是什么	93
5.8.2	再谈 If-Then	93
5.8.3	将 If-Then 和循环组合起来	94
5.8.4	Select Case 命令	94
5.9	在运行时操作对象	95
5.9.1	方法	95
5.9.2	代码中的 Font 属性	96
5.9.3	消息框和输入框	96
5.9.4	输入框	99
5.10	有多个窗体的项目	100
5.10.1	为多个窗体编写代码	100
5.10.2	在运行时怎样处理多窗体	101
5.10.3	在一个窗体中保持焦点(模态化)	102

5.11	程序示例:贷款利息计算器	102
第六章	信息显示与输出	106
6.1	在窗体上输出信息	106
6.1.1	示例:在窗体内居中文本	107
6.1.2	代码中的字体属性	107
6.1.3	显示列表数据	108
6.1.4	Tab 和 Spc 命令和分号	109
6.1.5	Format 命令	109
6.2	图片框	112
6.3	RichTextBox	114
6.3.1	RichTextBox 的属性	114
6.3.2	RichTextBox 用法举例	115
6.4	Printer 对象	118
6.4.1	打印机的有用属性和方法	119
6.4.2	Printers 集	120
6.4.3	打印 RichTextBox 中的信息	120
第七章	程序流程控制技术	122
7.1	重复操作	122
7.1.1	确定循环	122
7.1.2	不确定循环	128
7.2	条件判断	134
7.2.1	If-Then 块	136
7.2.2	它是什么	137
7.2.3	在消息框使用 If-Then	138
7.2.4	示例:Dir \$ 命令	138
7.2.5	If-Then 与循环组合	139
7.2.6	Select Case	139
7.2.7	用 If-Then 完成	141
7.2.8	示例:KeyPress 过程	142
7.2.9	示例:使用 QueryUnload 事件	143
7.2.10	条件编译	144
7.3	GoTo 语句	145
第八章	内置函数	146
8.1	字符串函数	146
8.1.1	分析字符串	146
8.1.2	Like 运算符和模糊搜索	153
8.1.3	其他字符串函数	154
8.2	Rnd(随机)函数	155
8.3	位操作	160
8.3.1	位级逻辑运算符	161
8.3.2	KeyUp 和 KeyDown	163
8.4	数学函数	165
8.5	日期和时间函数	167
8.5.1	Date 函数	167

8.5.2	数字日历函数	167
8.6	财务专用函数	170
8.6.1	其他财务函数	171
8.6.2	其余财务函数	173
8.7	常用函数	173
第九章	大型项目的建立:项目、过程和错误捕获	176
9.1	自定义函数和过程	176
9.1.1	Function 过程	176
9.1.2	函数的高级用法	179
9.1.3	子过程	187
9.2	过程和函数的高级用法:传引用和传值	192
9.3	标准(代码)模块:全局过程和全局变量	197
9.3.1	增加或删除代码模块	199
9.3.2	有关过程的作用域	199
9.4	Windows 接口函数	200
9.5	DoEvents 函数	201
9.6	错误捕获	202
9.7	关于程序设计的讨论	206
第十章	图形程序设计基础	208
10.1	图形基础	208
10.1.1	AutoRedraw 属性的一个特点	209
10.1.2	ClipControls 属性和 Paint 事件	209
10.1.3	再谈 Paint 事件	210
10.1.4	Refresh 方法	210
10.1.5	保存图形	210
10.2	屏幕测度单位、颜色和像素	211
10.2.1	屏幕测度单位	211
10.2.2	颜色	212
10.2.3	像素的控制	213
10.3	Line 和 Shape 控件	214
10.3.1	Shape 控件	214
10.3.2	Line 控件	216
10.4	线、框和圆	216
10.4.1	线	216
10.4.2	DrawWidth 和 DrawStyle	218
10.4.3	框	218
10.4.4	圆	219
10.4.5	椭圆和纵横比	220
10.5	PaintPicture 方法	220

参考文献

第一章 概 述

本章将对 Microsoft Visual Basic 4 进行简要介绍。我们假定读者没有使用 Visual Basic 的任何经验,或者甚至是没有使用任何一种程序设计语言的经验。

1.1 Visual Basic 对 Windows 的意义

图形用户界面(GUI)给微机产业注入了强大的生命力。图形的信息量是文字所不可比拟的。用户现在面对的是有着各种各样图标的桌面平台,并且包含有利用鼠标或菜单的各种各样的程序,它们取代了先前 DOS 用户已使用很久的(某些用户已经恐惧了很久的)C:> 提示符。

从长远的观点看,也许比 Windows 应用程序外观更为重要的是应用程序本身。Windows 应用程序通常有约定俗成的、一致的用户界面,这就使得用户有更多的时间关心应用程序本身,而不必为在哪个菜单或对话框中敲了哪个键以及做什么来花费更多的时间。

程序设计者对图形用户界面一向有着含糊不清的认识,但最初的用户喜欢图形用户界面,所以 Windows 的程序被希望是基于 GUI 模式的,因为它有着很好的外观并给人良好的感觉。因此,不论开发什么样的 Windows 应用程序,用户都将希望得到一个能高效地开发基于 GUI 的应用程序的工具。

很长时间以来,几乎没有开发 Windows 应用程序的工具。在 1991 年 Visual Basic 推出之前,开发 Windows 应用程序要比开发 DOS 应用程序困难得多。程序员需要考虑的事太多,诸如鼠标在干什么,用户在哪里,是否处在一个菜单中,用户是单击还是双击指定的位置。开发一个 Windows 应用程序需要专门的 C 程序员,而且完成最简单的任务也需要几百行的代码。即使是专家也会感到吃力(Microsoft Windows 的 SDK 软件开发工具包除了 C 编译器之外的资料就重达 7kg)。Visual Basic 极大地改变了人们对 Microsoft Windows 程序设计的看法以及使用 Windows 的方式。它是 90 年代完美的编程环境。Visual Basic 4 继续保持了这一传统:复杂的 Windows 95, Windows NT,甚至 Windows 3.1, Windows 3.11 以及 Windows for Workgroups 应用程序现在已经只需以前开发所需时间的一小部分就可以开发出来了。编程时也较少发生错误,即使有,也很容易检测和修改。在大多数情况下,用 Visual Basic 编写 Windows 程序已经变为一件既有效率又有趣的事了。

Visual Basic 允许用户将菜单、文本框、命令按钮、单选按钮(做出排他性的选择),复选框(做出非排他性的选择)、列表框、滚动条,以及文件和目录框增加到空白的窗口中。用户可以使用网格来处理表格数据,与其他 Windows 应用程序通信,以及访问数据库。用户可以在屏幕上显示多个窗口,这些窗口能访问剪贴板,并能访问同时运行的其他大多数 Windows 应用程序中的信息。用户可以使用 Visual Basic 来与 Windows 下运行的其他应用程序很方便地进行通信,这需要使用最新版本的 Microsoft 的 OLE 技术。

1.1.1 基本配置

Visual Basic 的标准版本需要在 Windows 95 或者 Windows NT 3.51(或以上版本)运行,至少需要一台可运行 32 位操作系统的高档微机。具体来说,用户需要:

(1) 一台高速的 486 或一台 Pentium 机,而且至少要有 8MB 的内存,有 16MB 内存则更理想。

(2) 至少应有 60MB 的空闲硬盘空间(如果希望把 Visual Basic 16 位和 32 位两个版本都装上,硬盘空间的需求就会超过 70MB)。

1.1.2 Visual Basic 4 的功能

Visual Basic 4 与先前三个版本相比具有以下优点:

(1) 不需要额外的工作就可以为 Windows 95 和 Windows NT 生成 32 位的应用程序(如果使用的是专业版或企业版,还可以从基本相同的代码中创建 16 位 Windows 3.x 应用程序)。

(2) 通过 Visual Basic,可以利用 Microsoft 的 OLE 技术——包括创建 OLE 服务器的能力。

(3) 可以创建使用某些面向对象技术编程方法的应用程序。

如果用户比较熟悉面向对象的编程方法(OOP),就会知道 Visual Basic 中的控件并不是 OOP 中所指的那种对象,并且 Visual Basic 也没有提供完全面向对象的开发平台。

(4) 可以扩展 Visual Basic 的编程环境,可以生成或使用第三方工具,这些工具可以很好地与 Visual Basic 环境结合在一起(例如,如果使用的是企业版,就可以内置 Microsoft 的 Source Safe 软件。Source Safe 软件可以通过实现版本的控制来简化 Visual Basic 在大型项目中的使用)。

(5) 可以使用条件编译来简化基于多平台的软件开发,参见第七章有关内容。

在 Visual Basic 下开发的应用程序能满足绝大多数情况下运行速度的要求(一般来说,在 Visual Basic 4 下开发的程序要比在 Visual Basic 3 下开发的稍慢一些,除非扩展了更多的内存)。这并不是说用户可以忽视用 C 或 C++ 来为 Windows 编程,现在,扩展 Visual Basic 的功能仍需要基于这些语言的工具来完成。

尽管 Visual Basic 企业版中的某些功能不是专业版中所包含的(本书中未作介绍),但它简化客户与服务器以及成组项目的开发过程。

1.2 基本操作

Visual Basic 有一个软盘版本以及一个 CD 版本(如果使用的是软盘版本,要做的第一件事就是把原来的安装盘做一个备份)。注意应把用户的注册卡寄出去,这意味着用户将会收到相当数量的邮寄宣传品,还能从 Microsoft 公司更容易地获得技术支持以及版本升级的通知。

1.2.1 运行安装程序

Visual Basic 的安装盘内包含着自动安装程序,可以安装不同版本的 Visual Basic。Windows 95 或 Windows NT 3.51 或者其后的版本必须运行 32 位的安装(Setup)程序。用户可以在 Windows 3.x 下运行 16 位的安装程序,可以运行无数次安装程序,这意味着用户不必受限于第一次所作出的选项(第一次安装时,最好以选择缺省方式进行安装)。第一次安装 Visual Basic 时,安装程序会提问用户的姓名及购买 Visual Basic 拷贝的公司名称。在以后每次运行 Visual Basic 时,将提示这些信息,让用户知道谁是该软件的合法使用者。

假定用户已经在 Windows 环境下安装了大多数的程序。从 CD 或第一张安装盘上找到 SETUP.EXE 程序,然后运行它。按照它的指示,不久就能将 Visual Basic 安装好,并且可以运行了(如果使用的是专业版或企业版,就需要与 32 位的版本分开安装,从而安装上 16 位的版本)。

如果在先前版本的 Visual Basic 目录下安装 Visual Basic 4,它将会覆盖掉先前的版本。因为 Visual Basic 4 可以通过最小的调整来运行使用 Visual Basic 1.0 或 2.0 所编写的程序,所以这通常没有问题。如果需要保留先前版本的 Visual Basic,那么应在与之不同的目录中安装 Visual Basic 4。

安装完成以后,Visual Basic 便出现在自己的程序组内。利用 Windows 95 的 Explorer 或 Windows 95 中标准的拖放技术,可以将 Visual Basic 移动到其他的程序组中或移动到工作台的上面,为此,可以参考随 Windows 所带的资料。

如果这些文档资料有所改动或增加,可参阅文件 README.TXT。如果这个文件分散在各张磁盘上,安装程序会自动将其拷入这个文件,并在 Visual Basic 程序组中为其加上一个图标,以便在 Windows 工作台上显示它。如果双击这个图标,Microsoft Windows 将会把这个文件加载到 WordPad 中。这样,阅读有关信息内容很方便。

用户不需要为 Visual Basic 安装专门的打印机驱动程序,Visual Basic 将使用 Windows 使用的所有的打印机信息。

1.2.2 运行 Visual Basic

要在 Windows 95 下运行 Visual Basic,最简单的方法是利用 Start 按钮上的 Program 选项(安装程序自动将它加到 Start 按钮上)。如果选择合适的 Visual Basic 的版本,此时会弹出一个子菜单,然后在孩子菜单上选择 Visual Basic 项目。当然还有多种其他方法来运行 Visual Basic:如移动到 Visual Basic 的目录中并双击 Visual Basic 的图标,从 Microsoft Windows 的桌面中启动 Visual Basic。还可以通过打开 Run 对话框并输入 Visual Basic 的路径,来运行 Visual Basic(用户在安装 Visual Basic 时可以自己设置其目录。缺省安装时使用的是 Windows 95 中 Program Files(程序文件)的“长目录名”,它在 DOS 窗口中为 PROGRA~1)。可以通过拖动来将它加入到 Start 菜单中,也可以为它生成一个快捷键(在 Windows 95 下这个列表会很长)。

当运行 Visual Basic 时,会看到屏幕上显示的有关版权信息,说明谁是该份 Visual Basic 拷贝的合法使用者。在此之后,将会出现如图 1.1 中所示的屏幕(第二章将详细解释该屏幕上的所有组成部分)。

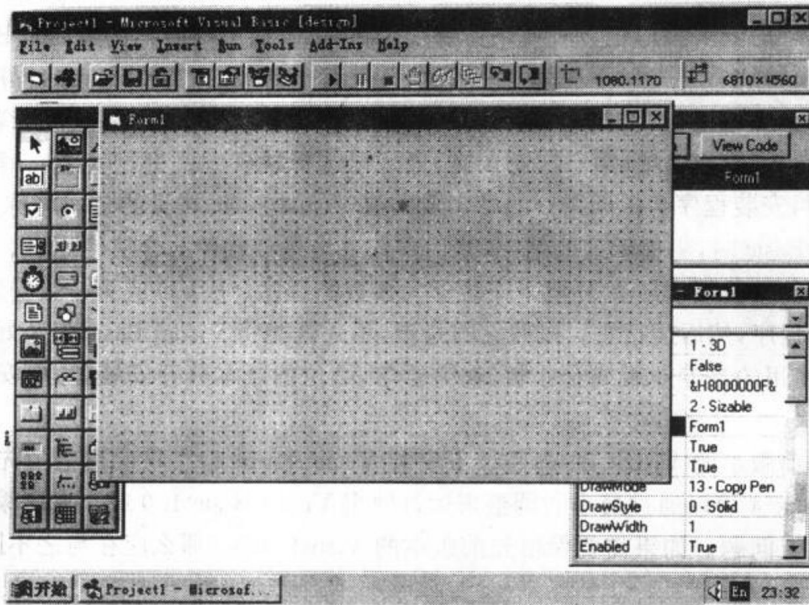


图 1.1 Visual Basic 的初始屏幕

1.2.3 如何使用 Visual Basic

在本小节中,用户将会学习如何创建最简单的 Windows 应用程序,即一个带有最大化、最小化以及退出按钮的尺寸可变的 Window(窗口)。尽管这个应用程序十分简单,但是执行下列步骤是用户使用 Visual Basic 的一个很有益的方法。

当运行 Visual Basic 时,看到的初始屏幕(如果使用的是 Windows 95)应与图 1.1 中所示的屏幕相类似(在 Windows NT 下会有一些细微的差别,用户屏幕的不同取决于用户所使用的 Visual Basic 的版本。正如前面已经讲过的,用户将在第二章中学习更多的有关 Visual Basic 环境的内容,所以现在只需跟着下面的步骤,而不必去担心什么环境)。

注意,图 1.1 中屏幕中间的空白窗口上有均匀分布的网格点。这就是用户要定制的窗体。用户可以使用网格来对齐控件,比如在屏幕上对齐命令按钮或列表框(第四章中将学习与之相关的更多内容)。当运行项目时(或经编译使它脱离 Visual Basic 开发环境而独立运行时),窗体就成了用户所看到的窗口。

在空白窗体的顶部是带有标题的标题栏。标题(Caption)是窗体标题栏上显示内容的 Visual Basic 术语。现在,该窗体的标题是 Form1,它是当用户启动一个新项目时,Visual Basic 为窗体提供的缺省标题名。

图 1.1 中 Form1 窗口的左边是工具箱。这里有各种可以放置在窗体上的控件。控件是 Visual Basic 中采用的术语,它是指用户设计的窗口中所使用的各种对象(用户将会在第四章中学习如何使用工具箱以及如何操作控件。可以使用的控件种类取决于所使用的 Visual Basic 的版本)。被部分遮盖住的是 Properties 窗口。用户可以利用它来定制窗体,以及各种各样的将要放置到窗体上的控件(用户将在第四章中学习如何使用 Properties 窗口)。

现在将注意力集中到名为 Form1 的中央窗口上。在许多 Visual Basic 应用程序中,完成设计时窗体的尺寸和位置就是运行该程序时所看到的尺寸和形状。这并不是说 Visual Basic 的基本特性就是能根据用户事件进行动态调整。

可以采用所有 Microsoft Windows 的应用程序都采用的方法来重新改变窗体的大小,这就是首先在窗体中单击来激活该窗体(用户可以根据标题栏醒目显示这一点来辨别窗口是否被激活)。然后再将鼠标移到窗体边框的任意位置。此时,用户处在热点(hot spot)中,鼠标光标会变为双箭头状。在该处,可以拖动窗体来改变它的尺寸和形状。同时,可以在标题栏中任意位置单击鼠标并拖动窗口到新的位置,从而移动窗体。窗体的尺寸和位置是 Visual Basic 里诸多被称为窗体属性中的两个例子。

要开发第一个示例应用程序,只要执行下列步骤:

(1) 通过在某些热点上操作或在屏幕上拖动窗体来改变窗体的缺省尺寸、形状以及位置。

(2) 按 F5 键或从 Run 菜单上选择 Start(按 ALT R,S 组合键或利用 Run 工具)来运行该项目。

注意,用户所看到的都是在设计时所留下的窗体,有着相同的尺寸、形状和位置的外观一般化的 Windows 窗口。当在 Windows 95 下运行该新项目时,弹出的窗口有着标准的 Windows 功能,比如大小尺寸可变的边框,控制框(在左上角),以及最大化、最小化和退出按钮(在右上角)。这显示了 Visual Basic 最重要的功能之一,即用户并未做什么事,但窗体已经成为用户所使用的 Windows 版本下具有各种功能的窗口!

要返回到开发环境,可以按 ALT+F4 组合键,或者双击控制框,也可以单击退出(exit)按钮。注意,用户的应用程序已经被自动赋予了关闭 Windows 应用程序的各种“标准”的方法。这表明,在许多情况下重要的一点是:不需要任何编程人员(或用户)的专门介入就可以使得 Visual Basic 的应用程序像 Windows 的用户所希望的那样运行。

1.2.4 运行 Visual Basic 项目

让我们通过运行一个随 Visual Basic 提供的示例程序来看看 Visual Basic 功能是如何的强大(Visual Basic 提供了数百个示例程序。当用户进一步熟悉 Visual Basic 时,就会发觉这些示例程序是例子与想法的很好资源)。这里要介绍的是如何运行一个简单的有四种功能的计算器。该项目只用了大约 75 行代码,但它提供了一个功能完整、面向可视的四功能计算器。

下面是加载现有项目的一般过程,项目是 Visual Basic 用来称呼根据某结构构成应用程序的文件,该结构是可视化的组件和代码的组合。

(1) 按下 ALT+F 组合键或单击菜单上的 File 选项来选择 File 菜单。

(2) 选择 Open Project 选项。

此时会打开一个如图 1.2 所示的对话框。

用户要使用的项目在子目录 SAMPLES 的子目录中(Visual Basic 提供的所有其他示例与之相同)。所以要移到 SAMPLES 子目录,然后单击标有 CALC 的目录,再双击名字为 CALC.VBP 的文件。

稍等片刻后,计算器项目文件将加载到 Visual Basic 中。按下 F5 来运行这个项目。当用

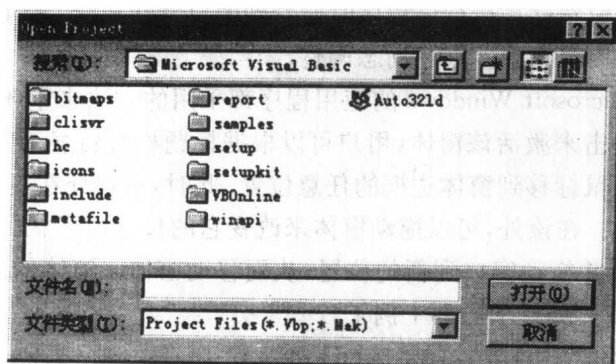


图 1.2 Open Project 窗口

户熟悉了该计算器后,可以单击它的退出按钮或按下 ALT+F4 组合键来返回到开发环境中。

1.2.5 联机帮助系统

建议用户现在至少要浏览一下联机教学系统的各个部分。这样做能补充或强化下面几章中所讲的内容。要使用联机教学系统,需要:

- (1) 在主菜单条上单击 Help 选项。
- (2) 再单击标有 Learning Microsoft Visual Basic 的选项。

当使用键盘快捷键来启动教学系统时,先需要按下 ALT+H 组合键,这将会打开 Help 菜单,然后再按下 L 键来选择教学系统。

假定已经安装了该教学系统,选择这些项目后通常会直接打开该系统。

教学系统非常易于使用。如果单击 Instructions 按钮,屏幕上会弹出 How to use the Learning Visual Basic Tutorial 消息框。当用户处在该屏幕状态下时,可以按下 ESC 键来关闭该弹出式窗口。教学系统只是通常的 Windows 95 帮助文件版本上的小调整,所以阅读它时不会碰上什么麻烦,没有必要按顺序学习这些教程,可以在任何需要的时候返回到该教程中(单击 Contents 按钮来返回到主屏幕)。教学系统一旦安装上,随时都可以利用它。

任何教程都可能对用户感兴趣或不感兴趣的东西,可以跳过不感兴趣的部分。可以双击标题来打开它,并将它从一个屏幕切换到另一个屏幕,这需要单击标题栏中标有 >> 的按钮。如果想返回到前一个屏幕,可以单击标有 << 的按钮。要返回到主屏幕并选择不同的教程,可以单击 Contents 按钮。

1.2.6 Visual Basic 应用程序开发概述

在本章结束之前,先对如何来开发 Visual Basic 应用程序的整个过程作个简单的概述。

开发 Visual Basic 应用程序的第一步是规划用户要看到的显示效果——换句话说,就是要设计屏幕。这包括使用什么样式的菜单? 应用程序所需的窗口尺寸有多大? 屏幕上显示几个窗口? 是否可以改变窗口的尺寸? 在哪里放置命令按钮使得用户可以单击它来激活该应用程序? 应用程序是否有地方(文本框)来输入文本?

就像先前已经提到过的,在 Visual Basic 中用户放置在正设计着的窗口中的对象被称为

控件。在进行工作时可以采用的控件数量取决于所使用的 Visual Basic 的版本。标准版的 Visual Basic 有多于 20 个的控件,专业版和企业版中可用控件多于 50 个。另外,因为 Visual Basic 已使第三方开发商为专门的工作任务开发了大量的控件,所以用户总可以为特殊的任务找到定制的控件。要生成自己的控件,一般要使用 C 或 C++ 语言来设计。

Visual Basic 与其他任何编程工具的不同之处在于它可以简化屏幕的设计。毫不夸张地讲,用户可以画出用户界面来,就像使用的是绘画程序。另外,当完成了界面的绘制之后,放置在空白窗口中的命令按钮、文本框以及其他的控件都可以自动识别用户的动作,比如鼠标的移动以及鼠标按钮的单击。Visual Basic 还具有菜单设计功能,它能够生成平常的那种菜单,也能生成快速弹出的弹出式菜单。

当完成了类似于传统的编程方法所要设计的界面后,就需要进行生成 Visual Basic 应用程序的第二个步骤:编写代码来激活用户在第一步中所创建的可视化界面。此时要求 Visual Basic 中的对象能够识别类似于鼠标单击这类事件,对象对这类事件的反应取决于用户编写的代码。用户几乎总是要为各个控件对事件的反应编写代码。这使得 Visual Basic 的编程方法与常规的编程方法有着本质上的不同。

常规编程语言所设计的程序总是从头向下执行。早期的编程语言都是从第一行开始执行,根据需要运行程序中不同的部分。而 Visual Basic 编程与之完全不同。Visual Basic 编程的核心是一组独立的代码块,它们只能被所设计的可识别事件来激活,并对该事件做出反应。这是一个本质上的转变。现在用户可以利用各种控件来代替设计一个程序以获得编程者想要的效果。

Visual Basic 中的大多数程序代码是表达用户如何去对诸如发生鼠标单击等事件作出反应,这类事件在 Visual Basic 中称为事件过程(event procedure)。实质上,Visual Basic 程序中可执行的每件事要么是一个事件过程,要么是一个事件过程被用来帮助其执行它的工作。第三步和第四步当然是查找代码中的错误(即调试),然后还要改正它们。

下面是用户要设计一个 Visual Basic 应用程序时应该采取的各个步骤的总结:

- (1) 定制用户所希望的窗口。
- (2) 确定窗口的控件都能识别哪些事件。
- (3) 为这些事件编写事件过程(还有子过程来满足事件过程工作的需要)。

下面是当应用程序运行时将会发生的一些事件:

(1) Visual Basic 监视着窗口和每个窗口中的控件,以及可以被每个控件都能识别的所有事件(比如鼠标的移动、单击、键盘的敲击等等)。

(2) 当 Visual Basic 检测到一个事件时,如果没有该事件的内置响应,Visual Basic 就会检查应用程序,以查看是否有为该事件编写的事件处理过程。

(3) 如果用户已编写了事件过程处理,Visual Basic 将执行该事件处理过程所对应的代码,并返回到步骤(1)。

(4) 如果用户并未编写相应的事件处理过程,Visual Basic 将会等待下一个事件,并返回到步骤(1)。

这些步骤连续循环执行,直至应用程序结束为止。通常情况下,在 Visual Basic 做任何事之前总有某些事件发生。这样事件驱动程序就会更多次地再激活——这使得程序对用户来讲更友好。