

Visual Basic 4.0 for Windows

Visual Basic 4.0 for Windows

# 高级程序设计

吴磊 程鹏 等编著

同济大学出版社

**Visual Basic 4.0 for Windows**

# 高级程序设计

吴磊 程鹏 等编

潘金贵 顾铁成 审校

同济大学出版社

## 内 容 提 要

本书讨论了 Visual Basic 4.0 for Windows 高级程序设计的方法和技巧,主要涉及用户环境、编程技术、对象、用户界面、调试技术、文件操作、动态链接、绘图操作、递归和数据库技术。本书内容丰富,实用性强,通俗易懂,相信有助于读者熟练掌握 Visual Basic 和进一步提高设计大型程序的能力。

对于希望快速掌握 Visual Basic 高级编程技术的读者,本书是一本实用教材,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 高级程序设计

吴 磊 程 鹏 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:17 字数:430 千字

1997 年 10 月第 1 版 1997 年 10 月第 1 次印刷

印数:1-3000 定价:22.00 元

ISBN 7-5608-1874-9/TP·194



# 前 言

本书介绍了目前最优秀的微机编程语言 Visual Basic 4.0 for Windows 的高级程序设计技术。由于 Visual Basic 4.0 的出现,使得计算机的编程变得更加简单、快速。本书选择了 Visual Basic 4.0 的精华,以简捷明了的方式介绍给读者。对环境、表单设计到程序设计都作了介绍,并探讨了用 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熟手,本书都会有很好的利用价值。

与前面三个版本的 Visual Basic 相比,Visual Basic 4.0 具有以下一些特点:

- (1) 能面向 Windows 95 和 Windows NT 生成 32 位应用程序。
- (2) 通过 Visual Basic,可以充分利用微软的 OLE 技术,包括建立 OLE 服务器。
- (3) 可以用面向对象编程技术创建程序。
- (4) 可以扩充 Visual Basic 的编程环境(可以在 Visual Basic 编程环境中无痕迹地创建或使用另外的工具)。
- (5) 条件编译使得开发多平台应用程序更加简单。

本书没有讲述怎样安装 Visual Basic 4.0,怎样启动它,以及 Windows 环境下的多窗口操作等内容。因为我们假定读者都已熟悉了这些内容。但是,我们并不认为读者具有 Visual Basic 或 Windows 编程方面的专门知识,而只是有在多种环境下编程的一般经验。

本书是一本关于 Visual Basic 4.0 编程的指南。本书的目的是提取 Visual Basic 4.0 的精华,把它们简单明了地介绍给读者。本书提出了用 Visual Basic 4.0 编程所必需的很多新的思考方法,所以,即使是传统编程语言方面的熟手,读了本书也会获益匪浅。如果继续用传统的思维来思考,就不能充分利用 Visual Basic 4.0 的功能。

根据读者的背景和需要,本书有多种用法。熟悉 Visual Basic 2.0/3.0 的读者可以快速浏览开始的几章,以便把更多的时间花在后面的章节上。因为这几章中包含有 Visual Basic 的底层部分。熟悉 Pascal 或 C 语言而不熟悉像 Visual Basic 这样的事件驱动语言的读者,也许想花更多的时间在前面几章,而只是略读关于 Visual Basic 底层程序的几章。通常情况下,建议读者认真阅读每一章,因为不管读者具有什么样的计算机语言背景,在每一章中都有关于 Visual Basic 4.0 的特殊信息。

参与本书编写的人员有:吴磊、程鹏、朱健、魏宁、程榆虹、李鸿、赵志鹏、常春华、陈辉、吴民韦、梁放。全书先由赵雅民教授和朱尽染研究员作了初审,最后由潘金贵和顾铁成审校定稿。此外,李纪鸿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

<b>第一章 Visual Basic 的基本环境介绍</b> .....	1
1.1 Visual Basic 4 的主屏幕 .....	1
1.1.1 标题栏概述 .....	1
1.1.2 菜单条概述 .....	2
1.1.3 工具条概述 .....	3
1.1.4 工具框概述 .....	4
1.1.5 初始 Form 窗口概述 .....	4
1.1.6 Project 窗口概述 .....	5
1.2 主要菜单概述 .....	5
1.2.1 Help 菜单概述 .....	5
1.2.2 File 菜单概述 .....	6
1.2.3 Edit 菜单概述 .....	8
1.2.4 View 菜单概述 .....	9
1.2.5 Insert 菜单概述 .....	9
1.2.6 Add-Ins 菜单概述 .....	10
1.3 如何装载和运行程序 .....	10
<b>第二章 高级编程基础</b> .....	12
2.1 数组的使用 .....	12
2.1.1 具有下标范围的列表 .....	13
2.1.2 使用多维数组 .....	14
2.1.3 使用 Erase 语句 .....	14
2.2 数据类型、指针和 GoTo 语句 .....	15
2.2.1 自定义类型 .....	16
2.2.2 使用模拟指针 .....	17
2.2.3 使用 GoTo 语句 .....	17
2.3 内部函数的使用 .....	18
2.3.1 按位逻辑操作符 .....	20
2.3.2 参数命名方法 .....	20
2.4 自定义函数和过程 .....	21
2.4.1 自定义函数 .....	21
2.4.2 定义 Sub 过程 .....	22
2.4.3 参数的按引用传递和按值传递 .....	22
2.4.4 在过程中使用列表或数组 .....	23
2.4.5 带有可选变量或参数个数的子程序 .....	24
2.5 使用递归 .....	24
2.6 创建较大项目的方法 .....	25
2.6.1 Standard(代码)模块 .....	26

2.6.2	DoEvents 函数的使用 .....	27
2.6.3	Sub Main .....	27
2.6.4	使用 Windows API 函数 .....	27
2.6.5	混合语言编程方法 .....	28
2.6.6	资源文件 .....	29
2.7	编译器的使用 .....	29
2.7.1	如何创建可执行文件 .....	30
2.7.2	条件编译技术介绍 .....	30
<b>第三章</b>	<b>对象设计 .....</b>	<b>32</b>
3.1	面向对象编程技术 .....	32
3.1.1	OOP 概念 .....	32
3.1.2	如何操纵 Visual Basic 中建立的对象 .....	34
3.1.3	由代码操纵对象变量 .....	36
3.2	集合 .....	39
3.2.1	用 Set 使用控件集合 .....	39
3.2.2	如何建立用户自己的集合 .....	40
3.2.3	Object Browser 对话框 .....	42
3.3	如何创建对象 .....	43
3.4	如何建立类 .....	45
3.4.1	创建新类模块 .....	46
3.4.2	一叠牌类模块示例 .....	46
<b>第四章</b>	<b>界面设计 .....</b>	<b>49</b>
4.1	使用工具箱 .....	49
4.1.1	使用框架 .....	50
4.1.2	使用选项(单选)按钮 .....	51
4.1.3	使用复选框 .....	52
4.1.4	使用列表框和组合框 .....	52
4.1.5	使用滚动条 .....	58
4.1.6	使用计时器 .....	60
4.1.7	设计菜单 .....	63
4.2	设计通用对话框 .....	69
4.2.1	样本文件浏览器示例 .....	69
4.2.2	使用通用对话框 .....	70
4.3	MDI 窗体及窗口设计 .....	75
4.3.1	MDI 窗体 .....	75
4.3.2	使用 Windows 菜单与 Arrange 方法 .....	76
4.3.3	与 Windows 有关的设计 .....	76
4.3.4	使窗体与调整尺寸和屏幕分辨率无关 .....	77
4.3.5	Visual Basic 的显示;ZOrder .....	80
4.4	专业版和企业版中的定制控件 .....	81
4.4.1	使用 Sheridan 3-D 定制控件 .....	82
4.4.2	使用 Desa Ware 活动按钮控件 .....	82
4.4.3	使用 Microsoft 通信控件 .....	83

4.4.4	使用 MicroHelp 计量(gauge)控件 .....	84
4.4.5	使用 MicroHelp 键状态控件 .....	86
4.4.6	使用 MAPI 控件 .....	86
4.4.7	使用 Microsoft 屏蔽编辑控件 .....	87
4.4.8	使用 Microsoft 多媒体控件 .....	88
4.4.9	使用大纲控件 .....	88
4.4.10	使用 Microsoft 图片剪切控件 .....	89
4.4.11	使用 OutRider 微调按钮控件 .....	90
4.4.12	使用 Sheridan 制表符对话控件 .....	91
4.5	使用 Windows 95 控件 .....	92
4.5.1	使用 ImageList 控件 .....	92
4.5.2	使用 ListView 控件 .....	93
4.5.3	ProgressBar 控件的使用 .....	94
4.5.4	使用 Slider 控件 .....	95
4.5.5	使用 StatusBar 控件 .....	95
4.5.6	使用 TabStrip 控件 .....	97
4.5.7	使用 Toolbar 控件 .....	97
4.5.8	使用 TreeView 控件 .....	99
4.6	使用 16 位 Windows 的工具条和状态条 .....	100
4.7	使用帮助系统 .....	100
4.7.1	如何写帮助主题 .....	101
4.7.2	设计上下文相关帮助 .....	104
4.7.3	如何建立和编译帮助文件 .....	104
4.8	访问 Windows 帮助机制 .....	105
<b>第五章</b>	<b>测试与调试 .....</b>	<b>108</b>
5.1	调试工具及测试方法 .....	108
5.1.1	调试工具及其功能 .....	108
5.1.2	测试程序的设计 .....	109
5.2	Debug 窗口 .....	110
5.2.1	Debug 窗口内的其他调试信息 .....	110
5.2.2	程序的单步执行 .....	111
5.3	程序设计应使测试更容易 .....	112
5.3.1	清除逻辑错误 .....	113
5.3.2	暂停程序执行 .....	113
5.3.3	利用断点进行测试 .....	114
5.4	程序调试技术综述 .....	118
5.4.1	错误循环的调试示例 .....	118
5.4.2	事件驱动错误和问题的调试 .....	118
5.4.3	文档和程序风格对调试的影响 .....	119
<b>第六章</b>	<b>文件操作 .....</b>	<b>121</b>
6.1	与文件有关的命令 .....	121
6.1.1	Shell 函数 .....	121
6.1.2	命令行提示信息 .....	122

6.1.3	与文件操作有关的函数 .....	123
6.2	使用文件系统控件 .....	124
6.2.1	使用文件列表框 .....	124
6.2.2	使用目录列表框 .....	125
6.2.3	使用驱动器列表框 .....	126
6.2.4	文件控件的综合利用 .....	126
6.3	顺序文件的操作 .....	127
6.3.1	使用 LOF 命令 .....	127
6.3.2	如何从一个文件中读信息 .....	129
6.3.3	如何向已存在的文件添加信息 .....	129
6.3.4	如何读取通用的顺序文件 .....	129
6.3.5	RichTextBox 控件和文件操作 .....	130
6.4	错误捕获 .....	130
6.5	随机存取文件、二进制文件及共享文件 .....	132
6.5.1	随机存取文件 .....	132
6.5.2	二进制文件 .....	133
6.5.3	在非文本模式下使用二进制存取技术 .....	134
6.5.4	共享文件操作 .....	135
6.5.5	Open 命令的一般格式 .....	136
<b>第七章</b>	<b>动态数据交换 .....</b>	<b>138</b>
7.1	使用剪贴板 .....	138
7.1.1	在 Visual Basic 中选择文本 .....	139
7.1.2	剪贴板格式和图形传输技术 .....	140
7.1.3	剪贴板程序示例 .....	141
7.2	如何激活 Windows 应用程序 .....	142
7.3	动态数据交换 (DDE) .....	145
7.3.1	建立 DDE 链接 .....	146
7.3.2	DDE 属性介绍 .....	147
7.3.3	处理 DDE 事件 .....	148
7.3.4	DDE 方法介绍 .....	150
7.4	利用 OLE 2 技术 .....	150
7.4.1	使用 OLE 方法 .....	152
7.4.2	建立 OLE 对象 .....	152
7.5	在设计时使用 OLE 2 .....	154
7.5.1	Paste Special 对话框 .....	155
7.5.2	OLE 属性介绍 .....	155
7.5.3	常用 OLE 容器方法 .....	156
7.6	OLE Automation 介绍 .....	158
7.7	OLE 服务器技术 .....	160
7.7.1	建立进程外 OLE 服务器 .....	160
7.7.2	建立进程内 OLE 服务器 .....	162
<b>第八章</b>	<b>绘图技术 .....</b>	<b>163</b>
8.1	图形操作基础 .....	164



8.1.1	AutoRedraw 属性的一个有趣功能 .....	165
8.1.2	ClipControls 属性和 Paint 事件 .....	165
8.1.3	Paint 事件的其他方面 .....	166
8.1.4	使用 Refresh 方法 .....	166
8.1.5	保存图片信息 .....	166
8.1.6	简单动画举例 .....	166
8.2	屏幕刻度 .....	167
8.2.1	定制刻度 .....	168
8.2.2	定制刻度的另一种方法 .....	170
8.3	使用 Line 和 Shape 控件 .....	170
8.3.1	使用 Shape 控件 .....	171
8.3.2	使用 Line 控件 .....	173
8.4	用代码画图 .....	173
8.4.1	使用颜色函数 .....	173
8.4.2	对像素进行控制 .....	174
8.4.3	示例程序:“Visual Basic A-Sketch” .....	176
8.5	线和框 .....	179
8.5.1	参考终点 .....	180
8.5.2	利用相对坐标 .....	181
8.5.3	使用网格图 .....	182
8.5.4	使用 DrawWidth 和 DrawStyle 函数 .....	183
8.5.5	设计方框 .....	185
8.5.6	动画和 DrawMode .....	190
8.6	圆、椭圆和饼图 .....	191
8.6.1	饼图示例 .....	193
8.6.2	椭圆和纵横比 .....	194
8.7	绘制曲线 .....	195
8.7.1	用简单公式描述的图形 .....	196
8.7.2	极坐标 .....	198
8.8	使用 PaintPicture 方法 .....	201
8.9	使用 Graph 控件 .....	202
<b>第九章</b>	<b>递归方法</b> .....	<b>209</b>
9.1	递归的基础知识 .....	209
9.1.1	使用递归函数 .....	210
9.1.2	简单递归过程 .....	213
9.1.3	汉诺(Hanoi)塔示例 .....	214
9.2	递归排序方法 .....	218
9.2.1	合并(Merge)排序方法 .....	219
9.2.2	快速排序(QuickSort)方法 .....	222
9.2.3	如何使排序稳定 .....	223
9.3	分形技术 .....	224
9.4	何时不用递归 .....	230
<b>第十章</b>	<b>数据库技术的使用</b> .....	<b>233</b>

10.1	现代数据库技术概述 .....	233
10.2	使用 Data Manager .....	235
10.2.1	使用 Data Manager 检查现存的数据库内容 .....	235
10.2.2	增加数据库内容 .....	237
10.2.3	使用 Data Manager 创建新数据库 .....	238
10.2.4	使用 Data Manager 的其他特性 .....	243
10.3	操纵数据控件 .....	245
10.4	利用数据控件编程 .....	247
10.4.1	Field 对象介绍 .....	249
10.4.2	数据控件的其他属性和事件 .....	250
10.4.3	关闭记录集合或数据库 .....	252
10.4.4	通过代码设置属性 .....	253
10.5	控制对数据库的修改 .....	254
10.6	SQL 语言简介 .....	256
10.6.1	使用 SELECT 语句 .....	257
10.6.2	使用 SQL 语句查找记录 .....	257
10.6.3	通过 SQL 语句修改表数据 .....	258
10.7	建立数据库对象 .....	258
10.7.1	DataBase 对象 .....	259
10.7.2	TableDef 对象和 TableDefs 集合 .....	259
10.7.3	Field 对象和 Fields 集合 .....	259
10.7.4	Index 对象和 Indexes 集合 .....	260

## 参考文献

# 第一章 Visual Basic 的基本环境介绍

本章将介绍怎样使用组成 Visual Basic 环境的菜单和窗口。读者将会理解 Visual Basic 中的那些联机帮助、编辑工具和文件柄实用程序。

注意,我们使用的是 Visual Basic 的专业版,若使用标准版或组织版,屏幕就稍有不同。

如果读者在视觉和感觉上还没有完全适应 Windows 应用程序,这章将给读者一个帮助。Visual Basic 本身就是一个设计得非常好的 Windows 程序,它的菜单和窗口操作也是 Windows 应用程序中非常典型的。熟练的 Windows 用户可以略读这些材料。但要记住,即使已完全熟悉了 Windows 应用程序,仍不足以完全利用 Visual Basic 的强大功能。

在 Visual Basic 应用程序的设计过程中建立窗口的缺省设置,使它很容易遵循 Windows 的标准。在 Windows 的缺省设置中,给了窗口适当的形状,并且可以根据用户的需要移动位置并重新定义尺寸。类似的,菜单也是按用户熟悉的方法来作出相应的反应。当然,Visual Basic 并不会把读者限制在这些缺省值内。但随意作出改动并不是一个好主意。

因为 Visual Basic 具有强大的功能、丰富的工具和详尽的菜单,所以一开始它就很容易掌握。为了减少混淆,在这一章中,将给出有关 Visual Basic 环境的详细叙述。

## 1.1 Visual Basic 4 的主屏幕

当启动 Visual Basic 时,将看到一个版权屏幕,上面指明了该软件的使用权属于谁。稍等一会儿,将自动地显示 Visual Basic 环境,如图 1.1 所示。

注意,Visual Basic 保存了上一次使用时屏幕上的最后的布局,并在下次启动时重新使用。所以,读者的屏幕看起来可能和图 1.1 不同。

屏幕上看起来可能会觉得很拥挤。标准的 Visual Basic 环境分为六个部分,但两个窗口都是暗的。Properties(属性)窗口用来定制一个窗体或控件。另一个窗口叫 Debug(调试)窗口,当运行或调试一个程序时,将在最上面显示该窗口。

下面是主窗口中其他部分的介绍,接下去的几节将顺序介绍菜单中最常用的项。

### 1.1.1 标题栏概述

标题栏是屏幕最上面的水平条,它指示了应用程序的名字,这一做法在所有 Windows 应用程序中是一致的。用户和标题栏之间的交互关系由 Windows 来处理,而不是由应用程序处理。在一个 Windows 应用程序中,标题栏和菜单条下的所有区叫用户区。读者设计的应用程序在外观上与读者在用户区中放置的对象完全对应。

在 Visual Basic 中,刚启动时标题条上总显示:

Project 1—Microsoft Visual Basic [design]

这在 Windows 应用程序中是很典型的:在复杂的程序中(比如 Visual Basic)有多种状态,标题栏的改变指示了相应的状态。例如,当在 Visual Basic 环境中运行一个程序时,标题栏将切

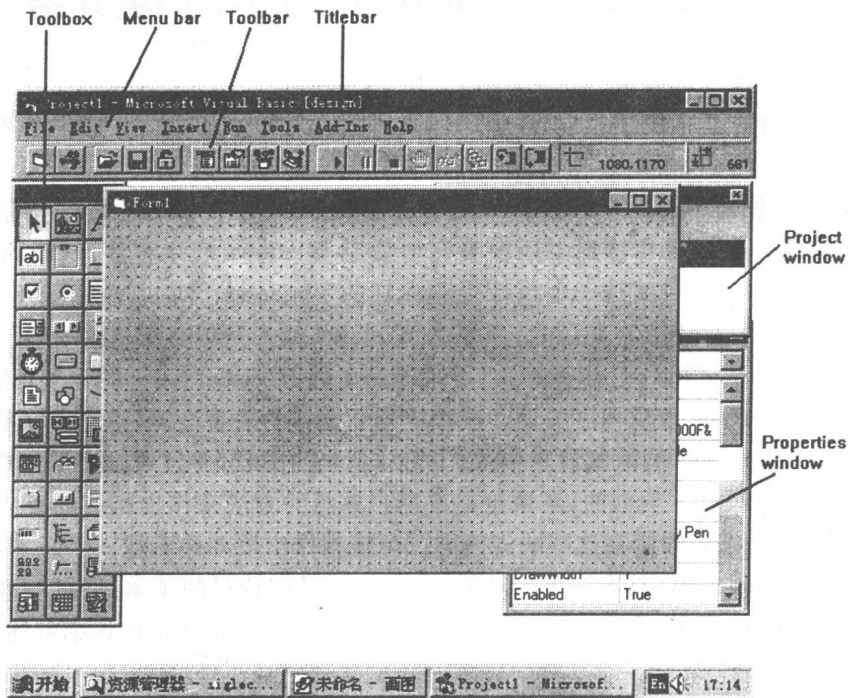


图 1.1 Visual Basic 4 环境

换为：

Project 1—Microsoft Visual Basic [run]

当调试(在程序中纠正错误)时,若暂时终止程序,标题栏则切换为:

Project 1—Microsoft Visual Basic [break]

### 1.1.2 菜单条概述

从菜单条中列出的下拉菜单中选择菜单项,是利用 Windows 应用程序的强大功能的常用方法之一。Visual Basic 本身也是如此。对于 Visual Basic 来说,菜单条中列出了在开发、测试和保存应用程序时所需的工具。File 菜单中包含了操作构成应用程序的文件的命令。Edit 菜单中包含了很编辑工具,可用它们来编辑代码,而这些代码是用来激活为应用程序设计的用户界面。其中包括 search-and-replace(查找与替换)编辑工具。View 菜单使得能对应用程序的各个部分快速访问。Insert 菜单用来插入新的对象。Run 菜单能让读者在开发应用程序过程中来检测它。Tools 菜单用来访问修改(调试)问题或故障的工具(第五章将详细介绍调试技术)。Add-Ins 菜单用来访问加到 Visual Basic 环境中的工具。最后,可以通过 Help 菜单来访问带有程序的联机帮助系统,或者运行它中间包含的最基础的提示。

注意所有的菜单中都有一个字母加了下划线。按下 ALT 键和带下划线的字母将打开这个菜单。另一种访问菜单的方法是单独按 ALT 键激活菜单条。当完成这项工作后,将会发现 File 菜单被高亮显示。这时可以用箭头键在菜单条上移动。按 ENTER 键或下箭头键将

打开该菜单。一旦打开了一个菜单,所有要做的工作是按快捷键(也叫热键),用来选择一个菜单选项。例如,如果打开了 Help 菜单,按 L 键则打开了提示。快捷键不是总起作用的。

有些菜单项具有快捷键。快捷键是一些键的组合,在不打开菜单的情况下,用户通过按快捷键就可以完成某一(些)操作。例如,Windows 应用程序有一个共同点,在不打开 File 菜单的情况下,按 Alt+F4 即可退出程序,对 Visual Basic 来说也是这样的。

### 1.1.3 工具条概述










工具条就在图 1.1 中的菜单条下,它是 Visual Basic 2.0 中加入的。这在 Windows 应用程序中越来越统一了。加入工具条图标是为了让用户不必要通过菜单就能激活一项任务。工具条上的每一项在键盘上都有一个等价键,而通过后者来选择很费事。

表 1.1 中列出了工具条中的工具,阅读时从左向右读。

表 1.1 缺省工具条图标

图 标	名 称	作 用
	新窗体	在项目中添加一种新窗体(可定制的窗口)。这个和 Insert Form 的功能一样
	新程序块	为特定的编程代码打开一个新程序块,这和 Insert Module 等价
	打开项目	可用来保存一个已经存在的 Visual Basic 工程(见 1.3 节)。它和 File Open 等价
	保存项目	用来保存 Visual Basic 项目,和 File Save 等价
	锁定控件	用来阻止一个非正常移动
	菜单编辑器	用来设计菜单。这和 Tools Menu Editor 等价,它有等效快捷键 CTRL+E
	属性	用来修改 Visual Basic 对象的大小、形状和颜色。这与快捷键 F4 或 View Properties 等价
	对象浏览器	用来打开对象浏览器对话框。这与 View Object Browser 或快捷键 F2 等价

(续表)

图 标	名 称	作 用
	项目	切换到 Project 窗口。这与 View Project 或快捷键 CTRL+R 等价
	开始	用来运行一个应用程序。在设计了一个应用程序以后,这与 Run Start 等价
	中断	暂停一个正在运行的程序,可以用 Run 工具或 SHIFT+F5 继续。这与 Run Break 或快捷键 CTRL+BREAK 等价
	结束	结束一个正运行的程序,与 Run End 等价
	设置断点	是一个调试工具。是在程序中的特定位置设置一个临时停止符。它来自于 Run 菜单,快捷键为 F9
	随时观察	另一个调试工具,将在第五章中介绍。在任一瞬态可对程序的多个部分进行观察。也来自于 Run 菜单。快捷键为 SHIFT+F9
	调用	显示当前调用的过程的列表。是一个调试工具。见第五章
	单步信息	在程序中一次只执行一行。也是一个调试工具。来自 Run 菜单,快捷键为 F8。将在第五章中讨论
	单步运行	因为在不同的过程中,由于不同的任务申请中断而使得程序变得越来越复杂时要用到它,见第五章。它一次也只执行一行,但对一个过程看作只是一步。也是一个调试工具。来自 Run 菜单,快捷键为 SHIFT+F8

#### 1.1.4 工具框概述

它在图 1.1 中所示屏幕的左边,工具条的下面。工具框中包含了开发应用程序的工具。有多少可用的工具取决于所使用的 Visual Basic 的版本,也取决于是否加入了定制控件。图 1.1 所示的工具来自于专业版的 Visual Basic。使用工具框可以在应用程序中放置命令按钮、文本按钮以及其他一些控件。

#### 1.1.5 初始 Form 窗口概述

初始的 Form 窗口在屏幕的中央。这是用来定制用户将看到的窗口。Visual Basic 说明书中为可定制的窗口使用的词为窗体。

注意,本书延用了 Visual Basic 手册中的一些约定,就是把可定制的窗口以及可以加到



窗口中的像命令按钮这样的对象叫窗体。

### 1.1.6 Project 窗口概述

为了使 Visual Basic 应用程序能共享代码或以前定制好的窗体, Visual Basic 把应用程序组织成项目文件。每一个项目中可以有多个窗体, 并且激活窗体中的控件的代码与窗体一起放在一个独立的文件中。被所有窗体分享的通用编程代码在应用程序中可以分为几个程序块, 每一个也可以独立存储。它位于图 1.1 所示屏幕的最右边。因为被 Form 窗口覆盖了, 所以 Project 窗口特别暗。在任一时刻, 可以通过选择 View|Project 或快捷键 CTRL+R 使 Project 窗口可见。在 Project 窗口中包含有一个由所有可定制的窗体和通用程序代码(程序块)组成的列表, 这些窗体和代码组成了应用程序。

在 Project 窗口中已列出了一项。这是要在其上创建应用程序的初始窗体。可以在查看窗体 (View Form) 按钮上单击鼠标, 使该窗体高亮显示并位于最上层。查看代码 (View Code) 按钮可以显示与该窗体相连的代码。

虽然 Visual Basic 把组成项目的所有文件单独保存, 但它总知道它们处在什么位置。它还创建了一个项目文件, 其中列出了组成项目的所有文件所处的位置。无论什么时候, 只要从 File 菜单中选择 Save Project 选项 (或从工具条中选择“保存项目 (Save Project)”工具), Visual Basic 就要创建 Project 文件。无论何时, 只要选择 Save Project As 选项, Visual Basic 就创建一个不同的 Project 文件。Project 文件名中都带扩展名. VBP。

注意, 对于 Visual Basic 4.0 以前的版本, Project 文件的扩展名为. MAK。

## 1.2 主要菜单概述

### 1.2.1 Help 菜单概述

联机帮助系统中包含了 Visual Basic 参考手册中的所有信息。另外, 它还有上百个示例程序和十几个很有用的表。

联机帮助系统有一个很有用的特点: 它随时可以获得与上下文有关的帮助。这意味着可以按 F1 或通过帮助菜单获得所需的帮助。可以获得有关 Visual Basic 编程语言中一些关键词的信息, 也可以获得有关错误的信息, 或者是 Visual Basic 环境的某一部分的信息。

一旦启动了帮助系统, 就可以把帮助窗口移到任一所需的位置。可以根据需要改变它的大小, 或把它缩小为一个图标。

下面是 Help 菜单中的每一选项的描述:

#### (1) Contents

可以用 ALT H,C (或者在 Help 菜单激活状态下按 C) 来激活 Contents 选项, 它将解释 Visual Basic 的帮助系统是怎样组织的。

#### (2) Search

可以用 ALT H,S (或在 Help 菜单已打开时按 S) 来激活 Search 选项, 它可以按指定的主题来查找帮助信息。

#### (3) Obtaining Technical Support

这个选项指示怎样从 Microsoft 获取技术支持。若 Help 菜单没有打开,可以按 ALT H, O 来激活它;若 Help 菜单已经打开,则可直接按 O 来激活它。它还包含了更进一步的主题的线索。

注意,如果读者是 CompuServe 的成员,通过 Microsoft 的 MSBASIC 论坛可以有效地得到自己所遇问题的解答。其他的用户可以记录这些问题,并可以经常获得很有帮助的建议。

#### (4) Visual Basic Books Online

这是 Visual Basic 中的一个新特点,它提供了 Book Online 工具,可以让读者阅读或查找 Visual Basic 的所有参考手册。

#### (5) Learning Microsoft Visual Basic

这是一个关于 Visual Basic 的简要信息。

#### (6) About

About 选项能告诉读者所用 Visual Basic 软件拷贝的序列号和版权,指出它的使用权属于谁。也能获得关于系统的一些信息。

### 1.2.2 File 菜单概述

读者需要一个主 File 菜单来操作组成项目的文件。这个菜单中包含有保存、装载和打印文件的命令。通过 File 菜单也能退出 Visual Basic。正如前面所见到的,当焦点位于主菜单条上时,可以按 ALT+F4 退出 Visual Basic,也可以像其他 Windows 应用程序一样,在主菜单条上打开控制框,然后选择 Close 选项;还可以在控制框上双击鼠标。

注意,在 Windows 95 下控制框是一个图标。

只有在开始开发应用程序时,才会用到主 File 菜单中的大多数选项,这在本章的后面要叙述。下面是关于这些项的一个简介,读者可以从中获得些帮助。

#### (1) New Project

New Project 不是装载当前的项目。如果在最后一次存储后,又对项目作了一些修改,将弹出提示对话框,询问是否要保存所做的工作。

如果按 ENTER 或 ALT+Y 回答 Yes,将弹出另一个对话框来保存文件。

#### (2) Open Project

Open Project 选项是用来操作一个已经存在的 Visual Basic 应用程序。正如该项后面的省略号一样,选择该选项将打开一个对话框。

#### (3) Save File

将当前活动对象保存到磁盘上。第一次选择该选项时,Visual Basic 将弹出一个像 Save File As 选项中一样的对话框。

#### (4) Save File As

选择该选项将弹出一个对话框,用来将当前活动的窗体或程序块(通用代码)保存到磁盘上,有可能用一个新名字。用该选项可以在不同的磁盘中保存该项目中某些块的备份,也可用来保存不同的版本。当当前应用程序的某一部分要用在别的项目中时,要用到该选项。这时还要用到 Add File 选项把该文件添加到不同的项目中。

#### (5) Save Project

该选项用来保存当前项目文件中包含的所有文件,并创建一个初始的 Project 文件。记

住,Project 文件是在项目中用到的所有的文件的列表,另外再加上 Visual Basic 要用到的一些信息。第一次选择该选项时,程序将打开一个对话框,该对话框和 Save As 选项中的对话框一样。

注意,在 Visual Basic 中,组成应用程序的窗体都按 ASCII 码格式存储,ASCII 格式对其他的应用程序来说是可读的。这叫只含文本的文件或叫文本文件。

#### (6) Save Project As

选择该选项将弹出一个对话框,让读者用一个新名字来保存组成当前项目的所有文件。它使用的方法是创建一个新的 Project 文件,并把组成项目的文件用当前文件名保存起来。也可以用这种方法来进行备份,把项目保存在不同的磁盘中或保存项目的不同版本。

#### (7) Add File

选择该选项将打开一个对话框,用来在应用程序中插入先前所作的工作。可以添加已完成的窗体、通用代码(程序块)、类代码块或者干脆是原文件。

#### (8) Remove File

该选项用来从当前所工作的 Visual Basic 应用程序中删除掉一部分内容。这个选项并不删除保存在磁盘上的文件。要删除磁盘中的文件需要用到 Windows 95 中的 Explorer(资源管理器)或者是使用 DOS 命令。

#### (9) Print Setup

该选项用来选择打印机,并设置打印纸类型和一些对象的选择。

#### (10) Print

该选项可用来打印当前的窗体、窗体中的代码、正在工作的程序块(代码),或者用来打印应用程序中的所有窗体和程序块。选择该选项将打开一个 Print 对话框。

其中的选项(单选按钮)包括 Selection、Current Module 和 Project,它们用来决定是要打印正在工作的应用程序中的一部分还是要打印整个应用程序。

Print 对话框中的三个复选框(Form Image、Form As Text 和 Code)用来决定是打印组成应用程序的所有或部分窗体,还是打印激活这些用户界面的代码。

#### (11) Make EXE File

选择该选项将打开一个对话框,用来创建 Visual Basic 应用程序,该程序能在 Windows 环境下独立于 Visual Basic 而运行。独立的 Visual Basic 应用程序需要动态链接库(DLL)文件以及可能的定制控件的文件。因为 Visual Basic 具有创建 OLE 应用程序的功能,所以最好使用 Setup Wizard,而没必要费劲地去决定应用程序需要哪些文件。

#### (12) Make OLE DLL File

将在第七章中进行讨论。

注意,这个菜单选项只在 Visual Basic 的专业(Professional)和企业(Enterprise)版中有效。

#### (13) The Most Recently Used(MRU) List

该选项记录了最近打开的四个 Visual Basic 项目。如果在表中列出的某个文件上单击鼠标,Visual Basic 将自动地装载这个项目。这使得很容易就返回到了以前的工作。

#### (14) Exit

选择 Exit 选项是退出 Visual Basic 的经常用到的方法。如果对当前的项目作了修改,Vi-