

VISUAL BASIC WORKSHOP

WINDOWS™ EDITION



Microsoft
VISUAL BASIC
WORKSHOP

编程训练

JOHN CLARK CRAIG 著
吴红艳 李宝东 译

Microsoft
P R E S S

学苑出版社

计算机语言技术系列丛书(二)

Visual Basic for Windows

编程训练

JOHN CLARK CRAIG 著

吴红艳 译

李宝东

熊可宜

学苑出版社

1994

(京)新登字 151 号

内 容 提 要

本书采用“示例学习”的方法,在开发和演示示范程序的同时,阐述了各种概念、技术和技巧,让读者按步骤阅读、修改实际程序,从而在有趣的示例学习过程中,高效地掌握和利用 Visual Basic for Windows 的编程技巧。

欲购本书的用户,请直接与北京 8721 信箱联系,邮编:100080,电话:2562329。

版 权 声 明

本书中文版版权由 Microsoft 授予北京希望电脑公司和学苑出版社独家出版、发行,未经出版者书面许可,本书的任何部分不得以任何形式或任何手段复制或传播。

Microsoft, MS, MS-DOS 和 Windows 是微软公司注册商标。

计算机语言技术系列丛书(二)

Visual Basic for Windows 编程训练

著 者:JOHN CLARK CRAIG
译 者:吴红艳 李宝东
审 校:熊可宜
责任编辑:甄国宪
出版发行:学苑出版社 邮政编码:100036
社 址:北京市海淀区万寿路西街 11 号
排 版:北京天奥科技公司激光照排中心
印 刷:兰空印刷厂
开 本:787×1092 1/16
印 张:24.75 字数:572 千字
印 数:1~10000 册
版 次:1994 年 7 月北京第 1 版第 1 次
ISBN7-5077-0905-1/TP·29
本册定价:59.00 元

学苑版图书印、装错误可随时退换

前 言

Visual Basic for Windows 3 是一个极好的程序设计环境。在 Visual Basic 已取得极大成功的情况下,Microsoft 仍投入极大的精力加以改进,升级到 3.0 版本。对有关老版本 Visual Basic 的一些缺点和局限性也得以平息。现在在功能、适应性和速度诸方面,Visual Basic 已能与 C 语言平起平坐。而在开发效率方面,Visual Basic 无疑占了上风。本书正是想给读者提供这方面的知识和思想,以便各位能很快就开始使用这种革命性的新方法,创建全功能的 Windows 应用程序。

作者希望各位能用 Visual Basic 愉快地工作。一般程序员通常要花几个月的时间才能用 C 语言熟练地开发 Windows 程序,但用 Visual Basic 则只需用几个小时熟悉这种语言,然后即可开发 Windows 程序。读者将十分惊喜地看到平生第一次开发 Windows 程序是那么简单、快速,完全超乎想象。

本书不像传统的教科书,而是让读者按照例子逐步学会 Visual Basic。文字部分是按开发和演示应用程序范例的步骤解释概念、技术和规律。Visual Basic 是一种交互性强、十分方便的程序设计语言,这样通过范例来学习语言就既有趣又有效。

本书选取的应用程序范例覆盖了一些作者认为每个人都会感兴趣的领域。读者也可以在具有 Visual Basic 程序设计的经验后,再扩充某些范例的功能。在此版中,我按照读者和 Microsoft 工作人员的建议作了一些修改。其中之一就是代码中的注释更多些。读者把随本书的软盘中代码装入 Visual Basic 环境时,彩色的注释行会使代码更容易理解一些。如果一个代码行太长,行尾将有一个特殊符号(<Enter>),表明这条语句持续到下一行。

我花费了相当多的时间改进一些应用程序的功能,使它们作为一个独立的模块时更有用一些;同时也演示了 Visual Basic 3.0 在 2.0 上新增的强大的功能。本书也添加了第一版上没有介绍的 Visual Basic 的其它功能。例如,第十章中的 COLORBAR 应用程序就演示了不在图形框而直接在窗体中绘制图形的方法,这也是调整显示器的对比度和亮度的一个简便的方法。本书中也提供了一些新的应用程序,演示了 Visual Basic 3 专业版的高级功能。

本书的初衷之一是提供一些窗体的源代码。以便读者可直接用在自己的应用程序中节省开发时间(在 Visual Basic 的术语中,窗体是指窗口和与之相关的控件、图标、图形及代码)。例如,第六章中列出的 GETFILE 窗体可以让用户从任何驱动器的任何目录中选择一个文件。这个窗体与几乎所有新的 Windows 应用程序中的 File Open 对话框类似。Visual Basic 3 中的通用对话框(Common Dialog)也提供了这些标准窗体。尽管如此,范例 GETFILE 中还是提供了源代码。读者可以随意修改,以适合自己的需要。

本书附带的软盘中包含了以下的章节中将要讨论的窗体和源代码(要了解更多的信息,可参阅简介后面的“如何使用本书附带的软盘”)。使用这张软盘可以节省键入程序的时间并避免敲键错误。

注意:读者如果已经熟悉了 Visual Basic 程序设计,就可以跳过本书的前面几章;如果熟悉 Basic 的早期版本而想学习 Visual Basic 增加的功能,可以研究一下第二章。

第一部分介绍了 Visual Basic for Windows 3 程序设计的入门知识。其中第一章简单回顾了 Basic 语言的历史,让读者了解 Visual Basic 是如何代表了一个进化的层面。第二章列举了 Visual Basic 的重要功能和优点,着重介绍了对它的最新改进。第三章通过一个非常简单的程序范例,比较了 Mricrosoft QuickBasic 和 Visual Basic 的语法解释。第四章引导读者逐步建立第一个 Visual Basic 应用程序。第五章则通过一个大的应用程序,逐步介绍了更多的 Visual Basic 程序设计概念。

第二部分提供了各种 Visual Basic 窗体和应用程序,读者可以根据自己的爱好选择某些专题。第六章提供的一些有用的窗体,可解决一般的程序设计问题,其中很多窗体还出现在本书的其它应用程序中。第七章可帮助读者了解更多有关 Visual Basic 内部运行知识。例如,一个程序显示所有的标准鼠标光标。另一个程序显示描述鼠标移动和按钮引起的鼠标事件的信息。第八章详细介绍了创建屏幕保护程序的方法,提供了现成的例子。第九至十二章提供了多种 Visual Basic 应用程序:一个高级计算器、日历、文件压缩和解压缩等等。

第三部分介绍 Visual Basic 高级程序设计功能。第十三章提供了有关动态数据交换 (DDE)的范例,介绍了 Visual Basic 应用程序与其它基于 Windows 的应用程序之间共享数据的一种方法。第十四章提供了有关对象连结和嵌入 (OLE)的简单但完整的范例。第十五章演示了新的数据控制方法,并介绍了如何操作数据库。第十六章介绍了如何调用 Windows 的 API (应用程序接口)功能。第十七章介绍如何书写并编译自己的动态链接库 (DLLs)。第十八章介绍了创建定做的控件的方法。第十九章提供了 Visual Basic 3 专业版中很多控件的操作范例。例如,对 NISTTIME 应用程序已作改进,可以使用新的 Communications 控件拨通美国标准及技术局 (NIST),校准机器时钟。其中最重要的一个新控件可以容易、高效地与多媒体设备通讯,我提供了一个应用程序范例,可搜索目录并播放 WAV 声音文件。

如何使用本书附带的软盘

本书附带的软盘中包含了书中介绍的所有 Visual Basic 编程设计和应用程序。无论读者是一个 Visual Basic 新手还是有经验的开发人员,这些范例和程序设计工具都可以帮助你更多地了解 Visual Basic for Windows 3。要了解这些程序创建、运行和重用的特定知识,读者可参阅本书的相关章节。

把附带软盘中的文件装入硬盘

1. 把此盘插入 A 驱动器中。
2. 在 MS-DOS 提示符后键入 a:install。
3. 按屏幕上的提示操作。稍等一会儿,软盘中的内容即被展开,并拷入 C 盘的 \WORKSHOP 目录下。

附带软盘中的内容

■ 本书中介绍的所有代码文件、窗体、消息文件和工作文件。其中包含超过 50 个以上的程序设计工作范例,覆盖从口令保护到动态数据交换 (DDE)的内容。

■ 每一个编程设计的可执行文件。读者只需启动 Windows 的 File Manager,然后在 EXE 文件上双击鼠标即可运行。

■ 所有动态链接库 (DLL)文件和定做的控件文件 (VBX)。用户没有 Visual Basic 时,运行 Visual Basic 3 应用程序必需这些文件。

本书附带的软盘可增加读者的兴趣,鼓励读者多做试验。运行软盘中的程序,读者可以观察各种特殊的效果;阅读本书的各章节,读者就可以了解这些效果是如何产生的。相信各位不久就可以创建属于自己的、更新颖的设计了!

目 录

第一部分 Visual Basic for Windows 入门

第一章 BASIC 简史	2
1.1 BASIC 的进展	2
1.2 Visual Basic	3
第二章 Visual Basic for Windows 的特点和优点	4
2.1 对 Quick Basic 的改进	4
2.2 Windows 的动态链接库	5
2.3 其它的 Windows 功能	6
2.4 未来展望	6
第三章 事件驱动的程序设计方法	7
第四章 第一个 Visual Basic for Windows 应用程序	9
4.1 WINDCHIL 应用程序	9
第五章 MOON 应用程序	22

第二部分 Visual Basic for Windows 的窗体与应用程序

第六章 常用的窗体	34
6.1 ABOUTDEM 应用程序	34
6.2 CONSTANT.TXT 文件	34
6.3 ABOUTDEM.MAK 文件	34
6.4 ABOUT.FRM 文件	35
6.5 ABORTDEM.FRM 文件	35
6.6 编辑 ABOUT 窗体	35
6.7 编辑 ABOUTDEM 窗体	39
6.8 FILEMSG 应用程序	41
6.9 GETDEMO 应用程序	48
6.10 GETTEST 窗体	49
6.11 GETFILE 窗体	51
6.12 SAVEFILE 应用程序	59
6.13 FILESHOW 应用程序	70
6.14 PASSWORD 应用程序	75
6.15 PROFILE 应用程序	79

6.16	PCTBAR 应用程序	85
6.17	NUMPAD 应用程序	90
6.18	EDITBOX 应用程序	97
第七章	开发 Visual Basic for Windows 应用程序	106
7.1	RGBHSV 应用程序	106
7.2	MOUSTPTR 应用程序	116
7.3	MOUSEVNT 应用程序	120
7.4	KEYEVNTS 应用程序	124
7.5	MSGBOXES 应用程序	128
7.6	INPUTBOX 应用程序	134
7.7	DRAGDROP 应用程序	137
7.8	MENUDEM 应用程序	141
第八章	屏幕保护程序	149
8.1	SSAVER1 应用程序	149
8.2	SSDOODLE 应用程序	153
第九章	日期和时间	164
9.1	VBCAL 应用程序	164
9.2	TWODATES 应用程序	176
9.3	VBCLOCK 应用程序	179
第十章	图形应用程序	191
10.1	COLORBAR 应用程序	191
10.2	PICRCLE 应用程序	193
10.3	FLOOD 应用程序	199
第十一章	随机数	204
11.1	RANDOMS 应用程序	204
11.2	LOTTERY 应用程序	213
第十二章	实用程序	220
12.1	FRACTION 应用程序	220
12.2	FILEFACT 应用程序	233
12.3	GRIDGRAF 应用程序	243
12.4	ICONVIEW 应用程序	252
12.5	IMAGVIEW 应用程序	257
12.6	FILECOMP 应用程序	264
12.7	LZ 应用程序	270

第三部分 高级编程概念

第十三章	动态数据交换 (DDE)	280
13.1	VBSOURCE 应用程序	280

13.2	VBDEST 应用程序	282
13.3	同 Word for Windows 进行动态数据交换	285
13.4	WORDSMIN 应用程序	285
第十四章	对象链接和嵌入 (OLE)	290
14.1	OLEDEMO 应用程序	290
第十五章	数据存取	294
15.1	Data 控件	294
15.2	约束控件	294
15.3	数据库对象	294
15.4	数据库	295
15.5	AREACODE 应用程序	295
第十六章	调用 Windows API 函数	302
16.1	FLASHWIN 应用程序	302
16.2	SYSINFO 应用程序	305
16.3	GRAFSAMP 应用程序	310
第十七章	建立你自己的动态链接库 (DLL)	316
17.1	C 语言源文件	316
17.2	BITS 应用程序	319
第十八章	创建定做的控件	325
18.1	VBINI 定做控件	325
18.2	MAKEFILE 文件	325
18.3	VBINI.DEF 文件	328
18.4	VBINT.RC 文件	328
18.5	VBINICU.BMP、VBINICD.BMP、VBINIMU.BMP 和 VBINIEU.BMP 文件	329
18.6	VBINI.H 文件	330
18.7	VBINI.C 文件	333
18.8	INIDEMO 应用程序	338
第十九章	定制的控件	342
19.1	DIALOGS 应用程序	342
19.2	THREED 应用程序	349
19.3	NISTTIME 应用程序	356

第一章 BASIC 简介

BASIC 编程语言由 John Kemeny 和 Thomas Kurtz 于 1963 年在 Dartmouth 学院创立。他们设计 BASIC 是为了将设计概念的教学，故以速度和效率为代价，着重程序的结构。其编写方法利用了作业控制语言 (JCL) 和 FORTRAN 和汇编语言中构造程序时所需的命令 (或称为宏)。实际上，它是一个编译过的程序。它负责解释用户输入的正文行。按条逐条读入，解释期执行一行程序 (或称为“解释”)。BASIC 允许用户在程序中编写任意复杂的算法和算法，而不必考虑计算机中的具体编程细节。

第一部分

Visual Basic for Windows 入门

BASIC 的早期版本具有许多独特的特征。程序的第一行以行号开头，行号通常不能省略 (更在，为提程序的可读性，语句通常缩进)。所有的字符均以大写字母输入和显示。用来控制程序流程的 GOTO 和 GOSUB 语句均以行号为目的地址。

这些特征鼓励了用户编写模块的“意大利面条式代码”，也就是说，程序的逻辑流程常常狂乱分支，就象杂耍的意大利面条碗在碟子上一样。图 1.1 给出了一些典型的意大利面条式代码。即使在这个简单的程序中，将控制流“丢”下去也是“迷路”。幸运的是，BASIC 从那时起已走过了相当的距离了。

早期的 BASIC 获得了其语言的美誉，它并不适合现实世界中的编程任务。不过，现在的 BASIC 已经从一种简单的、非结构的解释语言进化成一种快速的、结构化的编译语言。适合于建立各类应用程序。Hewlett-Packard, Microsoft 和其它软件公司已经推出了具有高级功能的 BASIC 增强版本。对于新入用 BASIC 的用户来说，现在是对该语言刮目相看的时候了！

```

10 REM-PRIME NUMBERS LESS THAN 100
20 N=1
30 IF N=100 THEN GOTO 120
40 I=1
50 I=I+1
60 I=N*I
70 IF INT(I)=I THEN GOTO 30
80 IF I>99(N) THEN GOTO 100
90 GOTO 50
100 PRINT N
110 GOTO 30
120 END

```



图 1.1 意大利面条式代码的早期 BASIC 程序

1.1 BASIC 的进展

BASIC 紧跟个人计算机的发展而发展。70 年代，Microsoft 率先为早期的个人计算机

第一章 BASIC 简史

BASIC 程序设计语言由 John Kemeny 和 Thomas Kurtz 于 1963 年在 Dartmouth 学院创立。他们设计 BASIC 是为了程序设计概念的教学,故以速度和效率为代价,着重程序的简洁性。其实现方法利用了作业控制语言以及在其他编程语言、如 FORTRAN 和汇编语言中构造程序时所需的编译/连接步骤(实际上,BASIC 系统本身就是一个编译过的程序。它负责解释用户输入的正文行。该系统每次读入、解释和执行一程序代码。因此,BASIC 被称为“解释语言”)。BASIC 允许用户集中精力考虑编程任务所需的方法和算法,而不必考虑计算机硬件的具体操作细节。

BASIC 的早期版本具有若干明显的特征。程序的每一行以行号开头,语句通常不缩格(现在,为提高程序的可读性,语句通常缩格)。所有的字符均以大写字母输入和显示。用来转移程序控制权的 GOTO 和 GOSUB 语句均以行号作为目的地。

这些特征鼓励了用户书写难读的“意大利面条式代码”。也就是说,程序的逻辑流程经常迂回和分支,就象柔软的意大利面条搁在碟子上一样。图 1.1 给出了一段典型的意大利面条式代码。即使在这个简短的程序中,沿控制流“走”下去也极易“迷路”。幸运的是,BASIC 从那时起已走过了相当的历程了。

早期的 BASIC 获得了玩具语言的名声,它并不适合现实世界中的编程任务。不过,现在的 BASIC 已经从一种慢速的、非结构化的解释语言进化成一种快速的,结构化的编译语言,适合于建立各类应用程序。Hewlett-Parckard,Microsoft 和其它许多公司已经推出了具有高级功能的 BASIC 增强版本。对于很久不用 BASIC 的用户来说,现在是对该语言刮目相看的时候了!

```
10 REM--PRIME NUMBERS LESS THAN 100
20 N=N+1
30 IF N=100 THEN GOTO 120
40 I=1
50 I=I+1
60 J=N/I
70 IF INT(J)=J THEN GOTO 20
80 IF I>SQR(N) THEN GOTO 100
90 GOTO 50
100 PRINT N,
110 GOTO 20
120 GND
```

图 1.1 意大利面条式代码的早期 BASIC 程序

1.1 BASIC 的进展

BASIC 紧随个人计算机的发展而发展。70 年代,Microsoft 率先为早期的个人计算机

引入了基于 ROM 的解释型 BASIC。例如, Radio Shack TRS-80 向公众介绍了 BASIC (和个人计算机的概念)。Microsoft BASIC 的这个原始版本至今仍以 GW-BASIC 的形式存在, 未经多少修改。4.01 版及更早的 MS-DOS 均提供了解释型的 GW-BASIC。

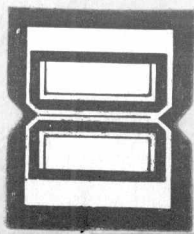
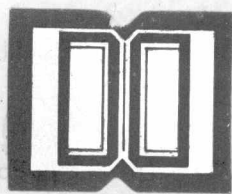
尽管 GW-BASIC 能够解决快速计算等简单问题, 但它更适合于充当玩具语言。没有一个软件开发者会想用 BASIC 来编写商用软件, 就象 MS-DOS 实用程序不会采纳批文件形式一样。原因是程序的执行速度很慢, 而且必须把源代码提供给用户。现在已经有了更好的程序开发方法。

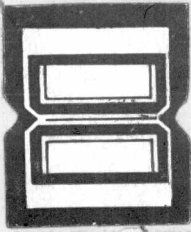
1982 年, Microsoft QuickBasic 革新了 Basic, 使之真正成为 MS-DOS 环境下的开发语言。QuickBasic 把 GW-BASIC 的交互性和多产性与编译语言的多功能和高速等特性有效地结合在一起。行号不再是必需的, 并增加了现代语言的某些特征 (如子程序和用户定义的结构数据类型)。QuickBasic 的图形和声音功能超过了 C、Pascal 和 Fortran。QuickBasic 程序还有一个优点: 程序既可以在交互式的解释模式下运行, 也可以编译成可独立执行的程序, 以供商用。

1.2 Visual Basic

当前正在进行的 Microsoft Windows 革命, 提供了功能强大而且是标准化的环境来更充分地发挥 Intel 公司最新微处理器的能力。对用户而言, Windows 是全新的: 它使得个人电脑更个人化并且更具友好性。但是对程序设计人员而言, 他们必须强迫自己又去学习一整套新的程序设计概念, 这样才有可能更有效地开发基于 Windows 的应用程序。但是, Visual Basic 改变了这种情形, 它是 Basic 语言的最新最大的一次改进, 现在学习给 Windows 开发应用程序已变成了一件激动人心的事情, 而程序设计工作也变得更具吸引力, 更有效率, 甚至更有趣!

过去的十年中, Basic 语言已改进了许多地方, 而 Visual Basic for Windows 3 也正是按这种趋势发展的。目前, 数以千计的基于 Windows 的应用程序是用 Visual Basic 的老版本开发成功的, 而以后更多的应用程序则会用最先进的 Visual Basic 3 开发。在下一章中, 我们将研究一下 Visual Basic 的一些独特的功能, 这样就会清楚到底它的妙处何在。





第二章 Visual Basic for Windows 的特点和优点

Visual Basic for Windows 的下列特点使其成为 Windows 下的理想开发语言。这些特点不仅加快了程序开发的速度,而且提供了开发复杂应用程序所需的全部工具。本章将着重介绍这些特点。

2.1 对 QuickBasic 的改进

Visual Basic for Window 保留了许多 Microsoft Quick Basic 的优点,同时针对基于 Windows 的应用程序的开发增加了许多功能。例如,用户可以很容易地把图形直接输出到窗口的任一部分,甚至在打印机上打印出来;可以从超过一千六百万种以上的颜色中选取图形对象的颜色(Windows 可以把像素混合提供近似的颜色,或者在硬件支持下生成一种颜色。无论哪种情形,用户都不必关心这些细节)。

相对于 Quick Basic, Visual Basic 所作的另一个改进是变量局部化的方式。这些规则容易理解和记忆,因为它们是经过改进的,而且说明得较详细。用 Visual Basic 开发的应用程序可以包含两种类型的文件:窗体,其中包含有关窗口和对话框的可见的代码;模块,其中只包含源代码。在一个窗体或模块中的通用定义段中定义的常量和变量,对于其中的任何一个子程序或函数都是可见的。在一个模块中定义为 Global 的变量或常量,对于本项目中的任一个窗体或模块都是可见的。除非在别处已定义成全局量,一般情况下,在子程序和函数中定义的变量和常量都局限于它们所在的例程中(等用户稍微有了一些 Visual Basic 的应用经验后,这些规则就显得更清楚了)。

图 2.1 列出了一个应用程序的代码。当用户在应用程序的窗口中按下鼠标按钮时,这个程序会显示下列信息:

```
3. 141593          6. 283186
```

注意:常量 PI 在本项目中任何地方都是可见的, TWOPI 在 Form 1 中任何地方也是可见的,但 THREEPI 只在子程序 Form_Load 中可见(从 Visual Basic 2 开始,未定义的变量的类型是 Variant 类型。在下面的例程 Form_Clide 中,THREEPI 被解释为一个空变量,也就是如果要显示它的值,将不会产生任何动作。在 Visual Basic 1 中,将显示 0)。

图 2.1 所示的代码看起来可能有些陌生,但不要着急,稍后我们还会讨论这个例子,到时自然就会明白。要记住的重点是我们可以根据变量和常量在项目中的位置,很容易地确定它们的可见性(可访问性)。

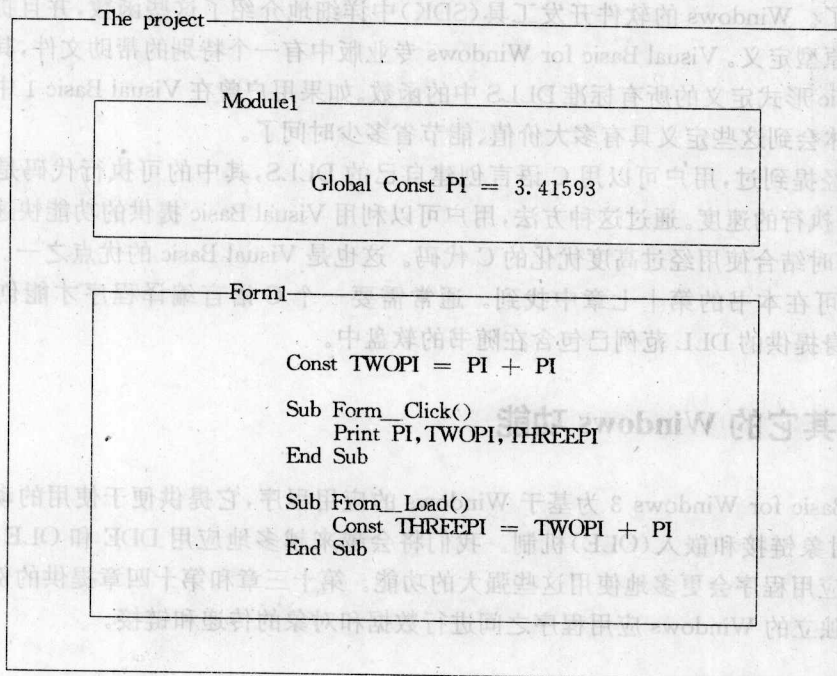


图 2.1 变量范围的例子

Visual Basic for Windows 的优点之一就是用户可以在短时间内成为一名多产的程序员。熟悉 QuickBasic 的用户很快会意识到 Visual Basic for Windows 的这个特点。用其它程序设计语言,如 C,开发基于 Windows 的应用程序时,用户不得不理解和掌握大量的编程方法,而这两种 Visual Basic 版本的交互性和直观性将使用户深受其益。目前,若想加快 Windows 应用程序的开发速度,最快速的方法就是学会用 Visual Basic 编程。

即使是经验丰富的 C 语言程序员也会欣赏 Visual Basic 的交互性界面的开发周期。在这一方面,它优于其它编程语言。例如,运行一两行 GW-BASIC 代码比在文档中找到相应的语句还要快速方便。现在,Visual Basic for Windows 利用这种交互开发概念,为应用程序创建复杂的用户界面。在界面的设计过程中,能够获得快速的视觉反馈是最为重要的。一般来说,很难把用户界面设计过程中的所有细节均记在心中,程序员将会发现,Visual Basic 的确节省时间。

2.2 Windows 的动态链接库

Visual Basic 的另外一个优点是它的可扩展性。虽然用 Visual Basic for Windows 开发的应用程序执行速度快,但在一些特定的领域中,经过优化的用 C 语言开发的程序执行速度更快。在本书的第二部分,用户可以看到一些例子,其中的 Visual Basic 应用程序调用 Windows 的 KERNEL、USER 和 GDI 动态链接库(DLLS)中用优化的 C 语言开发的 Windows 函数。DLLS 中有许多功能强大的函数,而用 Visual Basic 开发的应用程序可以很容易地调用它们。如果用户有 C 语言编译程序,也可以创建自己的有特殊用途的 DLLS。

如果用户已购买了 Windows, 就可以使用 DLLS 提供的功能, 因为它已集成在 Windows 操作系统中了。Windows 的软件开发工具 (SDK) 中详细地介绍了这些函数, 并且提供了供 C 语言调用的原型定义。Visual Basic for Windows 专业版中有一个特别的帮助文件, 其中包含了用 Visual Basic 形式定义的所有标准 DLLS 中的函数。如果用户曾在 Visual Basic 1 中调用过这些函数就会体会到这些定义具有多大价值、能节省多少时间了。

前面已经提到过, 用户可以用 C 语言创建自己的 DLLS, 其中的可执行代码是经过优化的, 可以提高执行的速度。通过这种方法, 用户可以利用 Visual Basic 提供的功能快速进行用户界面开发, 同时结合使用经过高度优化的 C 代码。这也是 Visual Basic 的优点之一。有关这种技术的例子可在本书的第十七章中找到。通常需要一个 C 语言编译程序才能创建自己的 DLLS, 但本身提供的 DLL 范例已包含在随书的软盘中。

2.3 其它的 Windows 功能

Visual Basic for Windows 3 为基于 Windows 的应用程序, 它提供便于使用的动态数据交换 (DDE)、对象链接和嵌入 (OLE) 机制。我们将会越来越多地应用 DDE 和 OLE, 因为基于 Windows 的应用程序会更多地使用这些强大的功能。第十三章和第十四章提供的例子演示了如何在各自独立的 Windows 应用程序之间进行数据和对象的传递和链接。

2.4 未来展望

现在用 Visual Basic 开发应用程序将使用相当长一段时间。Windows 操作系统设计得很稳定, 考虑到了将来计算机硬件功能的变化。例如, 计算机硬件的图像分辨率和颜色功能会增强, 但基于 Windows 的应用程序并不需要作任何修改, 它们只会运行得更快、更好。

第三章 事件驱动的程序设计方法

Visual Basic for Windows 程序是由事件驱动的,这个概念是现代程序设计语言的核心。程序员用过这类编程工具后,就不会再使用原来的编程方法。这种编程工具提供了一套崭新的、富有创造性和高效的技术。

幸运的是,事件驱动的代码与过程式代码在概念上的差异并没有人们开始想象的那么大。Visual Basic for Windows 程序的变化并不大。下面准备了一个小程序的三种版本,体现了从 GW-BASIC 到 QuickBasic,最后到 Visual Basic for Windows 的演化过程。Visual Basic for Windows 版本的扩充部分不多,但已经是一个事件驱动的程序了。

这个程序的三种版本在屏幕的顶端(Visual Basic for Windows 版本在窗口的顶端)显示 Testing.....1 2 3。先看看 GW-BASIC 版本(注意其中有老式的行号):

```
10 REM -- GW-BASIC Test Program
20 CLS
30 PRINT "Testing...";
40 FOR I=1 TO 3
50 PRINT I;
60 NEXT I
70 PRINT
80 END
```

这段程序的功能是清除屏幕,在显示器的顶端显示 Testing...,然后进入 FOR 循环,显示 1,2,3。

下面的 QuickBasic 程序显示同样的结果,从这个版本可以看出 QuickBasic 对解释型 Basic 的几个改进之处。

```
'QuickBasic Test Program
CLS
PRINT "Testing...";
FOR i= 1 TO 3
    Print i;
Next i
PRINT
END
```

现在,就轮到我们所要讨论的第一个全部由 Visual Basic for Windows 书写的、事件驱动的应用程序:

```
'Visual Basic for Windows Test Program
Sub Form_Click()
    Print "Testing..."
```



```
For i=1 To 3
    Print i;
Next i
End Sub
```

注意 Visual Basic 代码与 QuickBasic 代码是多么类似。关键字,例如 Print,第一个字母大写,其余字母小写,但语法不变。这种版本与老版本之间的主要差别是 Visual Basic 中已把代码放入一个名为 Form_Click 的事件驱动的子程序中。

注意:这个例子在 Visual Basic 1,2 或 3 中显示的内容是一样的,但在版本 2 和 3 中源代码是彩色显示的,所以操作起来更简单些。注释行是绿色的,关键字是蓝色的,而变量是黑色的。这只是 Microsoft 在对 Visual Basic 所作的两次重大修改中添加的许多漂亮的改进之一。

代码为什么不放在主程序中呢?这个问题的答案非常重要,它体现了 Visual Basic for Windows 编程思想的核心。的确,如果想保持与 QuickBasic 的兼容性,代码可以放在主模块中。但是,为了利用新的窗体、事件驱动代码,以及该语言的其它高级功能,包括与 Visual Basic for Windows 的兼容性,Visual Basic for Windows 程序的所有可执行代码必须放在子程序或函数中。在模块一级不应该有可执行的语句。这些子程序和函数由事件(如鼠标器的单击,按下命令按钮或选择正文框中的表项)驱动,或被其它过程调用(可以追加到一些事件驱动的例程中)。因为 Visual Basic 处理了所有日常琐事,应用程序所要做的就是对所发生的事件作出响应。

记住:Visual Basic 的所有可执行代码都必须放在一个子程序或函数块中!

事件驱动的子程序由 Visual Basic for MS-DOS 提供名字。在上面的测试程序中,Form_Click 表示,当用户在窗体窗口中单击鼠标器时,该子程序就运行。整个过程相当简单:运行程序时,先显示一个空的窗口,在窗口中的任一处单击鼠标器后,Testing.....1 2 3 显示在窗口中。

在这个例子中,Click(单击)事件激活了一个显示正文的子程序。同样,代码也可以放在 Form_DblClick 子程序中。当用户在窗体窗口中任一处双单击鼠标器按钮后,Testing.....1 2 3 就会出现。代码还可以放在 Form_Paint 子程序中,这些信息在程序运行后立即就会出现。

不难看出,在 Visual Basic for MS-DOS 中,还有许多事件可以用来激活子程序。从本书中的示范程序中还可以看出,事件驱动的程序设计方法是极其灵活的,非常适合于开发功能强,响应速度快的应用程序。