



华工松联电脑丛书

武汉松联公司
北京松岗

中文 Visual Basic 4.0 Win95

面向对象程序设计入门

林立域 策划 达米教室 编著



华中理工大学出版社

· 武汉松联环球电脑信息有限公司 · 北京松岗公司 ·

中 文

Visual Basic 4.0 Win 95

面向对象程序设计入门

林立域 策划
达米教室 编著

华中理工大学出版社

TP323.5

(鄂)新登字第 10 号

图书在版编目(CIP)数据

中文 Visual Basic 4.0 Win 95 面向对象程序设计入门/林立域策划 达米教室编著
武汉:华中理工大学出版社,1997.6

ISBN 7-5609-1527-2

I 中…

II ①林… ②达…

III 计算机程序语言-面向对象-程序设计

IV TP31

本书封面贴有华中理工大学出版社激光防伪标志,封底贴有台湾松岗公司防伪标志,无标志者不得销售。

版权所有 盗印必究

中文 Visual Basic 4.0 Win95 面向对象程序设计入门 © 林立域 策划
达米教室编著

责任编辑:汪厚祥等

责任校对:童兆丹

封面设计:梁书亭

监 印:张正林

出版发行者:华中理工大学出版社

(武汉市 邮编:430074)

经 销 者:华中理工大学出版社发行科

(电话:(027)7547012)

印 刷 者:武汉大学出版社印刷总厂

(邮编:430015)

(本书如有破损或装订错误,请寄回印刷厂更换)

开本:787×1092

1/16

印张:18.25

字数:420 000

版次:1997年6月第1版

印次:1997年6月第1次

印数:1-5 000

ISBN 7-5609-1527-2/TP·216

定价:33.00元

出 版 说 明

本书中文繁体字版由台湾松岗电脑图书资料股份有限公司（下简称“松岗公司”）出版。本书中文简体字版经松岗公司授权由华中理工大学出版社出版。任何单位或个人未经出版者书面允许不得用任何手段复制或抄袭本书内容。

由于海峡两岸计算机科学技术术语的译名不太相同，因此在出版中文简体字版时对正文和屏幕显示图上的术语进行了转译。转译工作是由汪厚祥、陈昊、马良荔完成的。转译内容力求做到表述准确贴切。因目前尚无简体字版软件故屏幕显示图仍为繁体字版，图上涉及的两岸译法不同的计算机科学技术术语已列出对照表附于书后，供读者查阅。

在中文简体字版中，对原中文繁体字版中某些仅适合台湾地区的内容经征得松岗公司驻北京代表叶权荣先生同意后作了删节，对原版书中一些错字、漏字也作了更正。原版书中附有练习实例磁盘，因其运行环境为中文繁体字版软件，与中文简体字版的正文内容不能对应，故中文简体字版均不带磁盘。

本书在中文简繁转译工作过程中得到了有关同志的大力帮助，谨此致衷心感谢。

华中理工大学出版社

1997年5月

内容简介

1991年 Microsoft 推出了 Visual Basic，对于开发 Windows 应用程序来说，它是一个可与 C 语言相当的、但较 C 语言容易使用的、功能强大的工具。最新版的 Visual Basic 4.0 又加入了很多新的特色，其中包括可以为 Windows 95 和 Windows NT 创建 32 位可执行程序。

本书以回顾 BASIC 语言的发展历程为序曲，徐徐拉开帷幕，向人们展示了一个穿上了新的 Windows 外衣重现舞台，光彩照人的新角色——Visual Basic。本书以大量的实例深入浅出地对 VB 环境，事件驱动程序设计，窗体控件、属性与方法，应用程序开发环境，输入与输出，变量与数组的定义，分支与循环；过程与模块，错误处理，窗体处理，按钮工具，OLE 应用，菜单编辑器，数据文件处理等都一一作了介绍。所用语言生动，所举例题实用。无论是对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。

目 录

第 0 章 Visual Basic 概述	(1)
0-1 Visual Basic 程序语言	(1)
0-2 BASIC 语言的历史	(1)
0-3 Windows 操作系统的来临	(2)
0-4 Visual Basic 入门简介	(2)
0-5 相关的名词及概念	(3)
0-5-1 对象	(3)
0-5-2 窗口	(3)
0-5-3 用户界面	(3)
0-5-4 事件	(3)
0-5-5 事件驱动程序设计语言	(4)
0-5-6 工具	(4)
0-5-7 工程	(4)
0-5-8 应用程序	(4)
0-5-9 可执行文件	(4)
0-5-10 自制品件	(4)
0-5-11 图标	(4)
0-6 工具的使用	(5)
0-7 事件驱动语言的使用	(5)
0-8 启动 Visual Basic	(5)
第 1 章 Visual Basic 透视	(6)
1-1 启动 Visual Basic	(6)
1-2 操作窗口	(9)
1-2-1 程序代码窗口	(10)
1-2-2 窗体窗口	(10)
1-2-3 属性窗口	(10)
1-2-4 控制窗口	(11)
1-2-5 工具箱	(12)
1-2-6 程序代码窗口	(14)
1-2-7 工程窗口	(15)
第 2 章 事件驱动程序设计	(19)
2-1 关于事件	(19)

2-2	事件过程	(20)
2-3	事件的驱动	(22)
2-4	鼠标事件	(22)
2-5	键盘事件	(24)
2-6	状态改变事件	(26)
2-7	链接事件	(27)
2-8	其他事件	(27)
2-9	概念复习	(27)
第3章	窗体、控件、属性与方法	(29)
3-1	设计程序的步骤	(29)
3-1-1	创建一个简单的实例	(29)
3-2	属性与方法	(32)
3-3	如何应用属性与方法	(32)
3-4	命名一个对象	(34)
第4章	应用程序开发环境	(36)
4-1	主菜单	(36)
4-2	文件菜单	(37)
4-2-1	与工程有关的处理	(37)
4-2-2	新增与移出文件	(38)
4-2-3	与打印有关的操作	(39)
4-2-4	产生执行文件选项	(39)
4-2-5	最近使用过的工程文件清单	(40)
4-3	编辑菜单	(40)
4-3-1	粘贴链接选项	(40)
4-3-2	查找与替换选项	(41)
4-3-3	凸排与缩排选项	(41)
4-3-4	移至顶层与移至底层选项	(42)
4-3-5	对齐格线选项	(42)
4-3-6	锁住控件选项	(42)
4-4	查看菜单	(43)
4-4-1	程序代码选项	(43)
4-4-2	窗体选项	(43)
4-4-3	过程定义选项	(44)
4-4-4	先前光标处选项	(44)
4-4-5	浏览对象选项	(44)
4-4-6	调试窗口选项	(44)
4-4-7	工程选项	(45)
4-4-8	属性选项	(45)
4-4-9	工具箱、工具条与调色板	(47)
4-5	插入菜单	(48)

4-5-1	过程选项	(48)
4-5-2	窗体选项	(48)
4-5-3	MDI 窗体选项	(49)
4-5-4	模块选项	(49)
4-5-5	类模块选项	(49)
4-5-6	文件选项	(49)
4-6	执行菜单	(50)
4-7	工具菜单	(50)
4-7-1	菜单编辑器	(50)
4-7-2	控件管理器	(51)
4-7-3	对象引用管理器选项	(52)
4-7-4	关于环境、工程、编辑器与高级选项	(52)
4-8	应用菜单	(54)
4-9	帮助菜单	(55)
第 5 章	简单的输入与输出	(57)
5-1	程序的输入	(57)
5-1-1	使用 InputBox 函数	(57)
5-2	程序的输出	(60)
5-2-1	使用 Print 方法	(60)
5-3	使用消息框	(65)
5-3-1	使用 MsgBox 函数	(65)
5-3-2	针对按钮定义的常量	(66)
5-3-3	对图标定义的常量	(67)
5-4	发出声音	(69)
第 6 章	变量与声明	(71)
6-1	变量	(71)
6-2	变量声明与类型规则	(72)
6-3	隐含与显式声明	(72)
6-4	建立一个变量范围	(73)
6-5	用户自定义数据类型	(74)
6-6	在程序中使用变量	(75)
6-7	使用局部变量	(76)
6-8	窗体层变量	(77)
6-9	模块层变量	(79)
第 7 章	数组变量的设置与应用	(82)
7-1	简单数组	(82)
7-2	数组的起始元素设置	(85)
7-3	指定数组范围	(87)

7-4	建立复杂的数组	(87)
第 8 章	分支与循环	(90)
8-1	控制结构	(90)
8-1-1	If/Then/Else/End If 结构	(90)
8-1-2	IIF 函数语法	(91)
8-1-3	Select/Case/Case Else/End Select 结构	(92)
8-2	循环结构	(92)
8-2-1	For/Next 循环结构	(93)
8-2-2	For Each/Next 循环结构	(94)
8-2-3	While/Wend 循环结构	(95)
8-2-4	Do/Loop 循环结构	(96)
第 9 章	过程与模块	(98)
9-1	区分模块与过程	(98)
9-2	通用与事件过程	(99)
9-3	过程	(99)
9-4	在过程中用参数传递数据	(100)
9-5	函数	(101)
9-6	过程层次	(103)
9-7	建立递归过程	(104)
9-8	过程范围	(105)
9-9	范围与可见度	(105)
第 10 章	错误处理	(107)
10-1	错误处理指令	(107)
10-2	建立错处处理过程	(108)
10-3	调试系统	(113)
10-4	使用断点	(114)
10-5	指定执行方式	(115)
10-6	使用调用窗口	(116)
10-7	调试窗口	(117)
10-8	使用调试对象的输出方法	(118)
10-9	使用监视功能	(119)
第 11 章	浏览对象	(122)
11-1	什么是浏览对象	(122)
11-2	对象引用类型	(122)
11-3	浏览对象对话框	(123)
11-4	对象引用管理器	(125)

第 12 章	窗体处理	(127)
12-1	建立一个窗体	(127)
12-2	移动窗体的位置	(127)
12-3	修改窗体的大小	(128)
12-4	修改窗体的边框格式	(129)
12-5	给窗体加上背景颜色	(131)
12-6	在窗体上写字	(132)
12-7	窗体上的显示层次	(133)
12-8	改变正文属性	(134)
12-9	加上背景图像	(135)
12-10	改变最小化后的图标	(136)
12-11	其他窗体属性	(138)
12-12	建立 MDI 窗体	(139)
第 13 章	按钮工具	(143)
13-1	指令按钮工具	(143)
13-2	选择框	(147)
13-3	框架与选项按钮	(151)
第 14 章	文本框与标签	(155)
14-1	文本框	(155)
14-2	标签	(156)
14-3	和其他控件的应用	(157)
第 15 章	滚动条与旋钮	(158)
15-1	滚动条	(158)
15-2	关于滚动条属性	(160)
15-3	用滚动条控制大型文件的显示	(161)
15-4	使用滚动条来显示运行的过程	(163)
15-5	旋钮	(164)
第 16 章	计时器	(167)
16-1	计时工具	(167)
16-2	使用计时器工具设计电子钟	(168)
16-3	控制图片的显示与消失	(170)
16-4	开启与关掉计时器	(173)
第 17 章	列表框	(176)
17-1	列表框的类型	(176)

17-2	建立一个列表框	(176)
17-3	选择多个项	(178)
17-4	下拉式列表框	(181)
17-5	驱动器列表框	(183)
17-6	目录列表框	(184)
17-7	文件列表框	(184)
第 18 章	数据与表格	(187)
18-1	数据工具	(187)
18-2	表格工具	(190)
第 19 章	OLE 应用	(196)
19-1	链接与嵌入的定义	(196)
19-2	OLE 容器	(196)
19-3	增加 OLE 控件到工具箱	(203)
19-4	删除一个 OLE 控件工具	(206)
第 20 章	通用型对话框	(208)
20-1	CommonDialog 控件	(208)
20-2	打开旧文件对话框	(209)
20-3	另存新文件对话框	(212)
20-4	颜色对话框	(214)
20-5	字体对话框	(215)
20-6	打印对话框	(217)
20-7	调用桌面上的帮助	(219)
第 21 章	绘图应用	(221)
21-1	Visual Basic 的绘图工具	(221)
21-2	直线	(221)
21-3	形状	(223)
21-4	图像	(227)
21-5	图片框	(228)
第 22 章	菜单编辑器	(230)
22-1	菜单控件	(230)
22-2	制作菜单的过程	(230)
22-3	加上热键	(236)
22-4	快捷键	(237)
22-5	其他重要的设定	(239)

22-6	建立子菜单	(240)
第 23 章	自定义数据结构	(243)
23-1	简单结构	(243)
23-2	嵌套数据结构	(245)
23-3	使用复杂的结构	(246)
第 24 章	数据文件处理	(256)
24-1	数据文件的输入和输出	(256)
24-2	Open 语句	(256)
24-3	FreeFile 函数	(258)
24-4	Close 语句	(259)
24-5	顺序文件	(259)
24-6	Print # 语句	(260)
24-7	Write # 语句	(261)
24-8	Input # 语句	(262)
24-9	Line Input # 语句	(264)
24-10	Eof 函数	(265)
24-11	随机文件	(265)
24-11-1	记录	(266)
24-11-2	有关随机文件的语句格式	(267)
24-11-3	打开随机文件	(268)
24-11-4	设定记录字段长度	(268)
24-11-5	关闭随机文件	(268)
24-12	随机文件的存取过程	(269)
24-12-1	Put 语句	(269)
24-12-2	Get 语句	(270)
24-12-3	使用 Seek 与 Loc	(270)
24-12-4	使用 Eof 函数	(272)
24-13	二进制文件	(272)
24-14	使用二进制文件的步骤	(272)
24-14-1	二进制文件的语句格式	(273)
24-14-2	Open 与 Close 语句	(273)
24-14-3	Get 与 Put 语句	(273)
24-14-4	使用 Seek 语句	(274)
24-14-5	使用 Seek 与 Loc 函数	(274)
24-14-6	使用 Eof 函数	(275)
24-14-7	二进制文件实例说明	(275)
24-15	文件处理实例	(275)
附录	简繁体版有差异的计算机常用名词术语对照表	(280)

90 年代初期, 正统的程序语言为过程化语言 (Procedural Languages), 例如 BASIC 及 C 语言, 然而这些语言现已逐渐衰落。取而代之的是面向对象语言 (Object - Oriented Languages)。例如 C++ 语言。然而旧景重现, 无人能明确地预见未来的新兴语言是什么。

当成千上万程序设计员急匆匆地改学 C++ 语言时, 有一小群人仍持续地置身于 BASIC 语言的使用及改进。当操作系统由 DOS 转移至 Windows 系统时, 适用于 Windows 操作环境发展的新 Visual Basic 语言诞生了。

0-1 Visual Basic 程序语言

Visual Basic 语言相对于其它 Windows 环境开发工具, 较易学习及使用。开发同一个程序时, 使用 Visual Basic 语言可较 C 或 C++ 语言省一半的时间。

“对象” (Object) 两字在电脑专用术语中已被滥用。例如 C++ 语言被认为是一种面向对象语言, C++ 语言的追随者报怨 Visual Basic 语言不算是面向对象语言, 因为它并不真正的符合面向对象应具有的特征。

事实上, Visual Basic 语言并非是要创造一个新的类别来定位其为一个面向类语言 (Class-Oriented Languages), 而是定位其为一个面向对象语言。Visual Basic 语言准许你创建一个类别, 并可定义专用属性及方法。

此外, 销售 Visual Basic 扩展对象的行业正在兴起, 这些自制控件 (通常又称为扩展控制 VBX,) 提供某些特定的功能, 而大约有 50% 的 Visual Basic 程序员对此大感兴趣。程序员可依自己作业所需, 利用某些符合需求的代码来解决问题。

这些由专家专门为程序设计员所需而编写的控件, 是一种可重复使用的控件。而此概念最终证明是来源于 Visual Basic 语言, 在 C++ 语言中则设有此种概念。

而在 Visual Basic 4.0 中, Microsoft 更扩充了这种自制控件, 并给予其一个新的名称 OCX (对象链接及嵌入控件) 代表 OLE Control Extension。功能上而言, OCX 酷似 VBX。

0-2 BASIC 语言的历史

Visual Basic 语言部分来源于 Beginners All purpose Symbolic Instruction Code (BASIC)。BASIC 语言是在 60 年代初期于 Dartmouth College 发明出的。BASIC 语言的发明, 主要是想使程序设计易于学习。

在 50 年代程序语言主要是用于解决数学问题。当时主要使用电脑的人员是科学家及工程师, 对他们而言, 使用那些复杂的程序语言都不成问题。当步入 60 年代时, 学术界意识到电脑将会大有用途, 而不再局限一方。他们极力想推广电脑的用途, 却因为繁杂累赘的程序语

言而困扰不已。

早期于 Dartmouth College 所发明出的 BASIC 语言即是在此背景下诞生的。初期的 BASIC 语言是一种易学易用的语言。在过去的 25 年中，大部分的电脑科学家所学的第一种语言就是 BASIC 语言。

在 70 年代，微处理器技术已进步到可制造娱乐用小型电脑。在 1975 年第一部台式电脑 MITS Altair 正式上市。它有 4k 的 RAM 可执行以组合语言所写的程序。当时 Bill Gates 及 Paul Allen 写了一套 Altair 版的 BASIC 语言。这使得 BASIC 语言成为被各处的个人电脑使用者所采用的制式语言。

整个 70 年代，几乎所有的程序都是采用 BASIC 语言编写的。在 1982 年，当 IBM PC 问市时，其 DOS 操作系统已内带有 BASIC 语言（GW-BASIC 版本）。

随着程序语言的改进，Microsoft 公司加强了 BASIC 语言的性能，并改名为 QuickBasic 语言。QuickBasic 语言是专利产品，它有一个支持结构化及图形化操作的调试环境。在开发 DOS 应用程序时，它是一个十分有用的工具。现今大部分在 DOS 下使用的 QuickBasic 都被称为 QBASIC。

0-3 Windows 操作系统的来临

大约又经过了 10 年，Microsoft 推出了 Windows 3.1 系统。在当时，DOS 系统程序员对它并不感兴趣。但用户对其却是印象深刻。

Windows 3.1 的舒适的图形外观及直接式界面，很快地使其成操作平台的最佳选择。突然间，即使是最棒的 DOS 应用程序，亦令用户感到不满。他们想使用更多的字体、同时使用多个应用程序、对话框、下拉式菜单等许多非常“酷”的特点。而这些都是 DOS 系统应用程序无法实现的。

当使用者对 Windows 应用程序的到来鼓掌不已时，程序员却开始感到头痛万分。程序员原有很大的自由决定程序中的一些事件如何处理。但是在 Windows 系统下，一切事件都得遵守一堆严格的规定来处理。这使得他们觉得 Windows 系统有如洪水猛兽般的可怕！坦白而言，这确实是难题。

程序员除了得熟悉 Windows 及 BASIC 指令外，还得学习新的 C 语言及成百的函数调用。更糟的是，每一个函数调用都得加入 5~6 个参数。以最简单的 4~6 行的 BASIC 代码，换作是 Windows 程序码，需要写好几页指令。这是千真万确但令人感到厌烦的事实。

0-4 Visual Basic 入门简介

正当 BASIC 语言逐渐消逝且将要被人们埋葬之际，它穿上了一件新的窗口外衣（Microsoft Visual Basic）重现舞台，而并依此重辟一条康庄大道。在表面上，它似乎没有什么重大的进步，乍看之下也似乎没有什么特别的事发生过一般，但在市场上却获得了出人意料的成功。

Visual Basic 成功的原因有几个：第一个原因是在大部分的应用程序中，它不用写传统的窗口程序。第二个原因是它以简单的 BASIC 语言代码取代了 C 语言代码。如此一来，使得业余的程序设计人员，也可加入 Windows 程序开发的行列中。

第三个原因是，专业的程序员可使用 Visual Basic 语言快速地开发应用程序中较简易的部分。因它可直接调用储存在动态链接库中的函数 (DLL)，程序员可用 C/C++ 语言开发程序中较困难的部分，并将它储存在一个 DLL 中，再在所设计的 Visual Basic 应用程序中使用它。这一点使得 Visual Basic 语言十分符合专业程序设计的需求。

第四个原因是专营由非主流软件公司所生产的扩展对象销售商有所增加。事实上，在市场上已有数百种的扩展对象在销售。这些产品看起来就好像是 Visual Basic 自带的位图按钮，且功能上也像 Visual Basic 本身的功能一样。

在 Visual Basic 4.0 中，Microsoft 公司强调已将 VBXs 转换成对象链接及嵌入控件 (OLE Control Extensions, OCX)。功能上而言 OCX 酷似 VBX，但它们都与其它语言不相干。而这些丰富的控件使得 Visual Basic 语言大为畅销且更增其实用性。在本书中，你将会使用到几个扩展对象，但如何写一扩展对象，却非本书所论及之范围。

Visual Basic 语言受到专业程序员喜爱的第五原因是，不须付发售版权费用。也就是说当你写出一个很有用处的程序且你想将这个程序分售给他人时，毋须付费给 Microsoft 公司或购买特别的发售工具。

0-5 相关的名词及概念

想要在 Windows 程序设计中成功，那了解它的术语是件重要的事。许多用于描述窗口概念的名词，在其它的地方有不同的意义。在这一节中，将列举出几个这样的名词。

0-5-1 对象

对象 (Object) 是一个具有属性及特定动作的软件部分。对象的动作与数据类型十分类似但更为复杂。我们可通过属性 (Property)、方法 (Method) 及事件 (Event) 来使用对象。

属性是对象的变量及常量。方法是与对象相关的命令及函数。每一个你想要实现的操作，对象都有个相对应的方法来实现。事件是专门控制对象的中断及操作。按鼠标这个操作就是一个事件的例子。

在 Visual Basic 语言中，对象可以是窗体、打印机、窗口、按钮、图片或是动画等等。

0-5-2 窗口

窗口 (Window) 是一个在屏幕上包含整个或是部分应用程序的长方形区域。窗口可视为桌面上的一张纸。你可把一个窗口叠在另一个窗口之上，也可将它摆在任何位置，还可以在上面读写。

0-5-3 用户界面

用户界面 (User Interface) 是电脑程序使用中真实可见的部分。包含了可供用户控制的部分，如按钮、窗口、图形、声音控制等等。

0-5-4 事件

应用程序所激发的活动就是事件 (Event)。通常这些事件的启动是依用户的响应形成的。但事件的启动亦可由像是“时间超过期限”或是程序控制等因素造成的。例如，窗体装入、数

据输入等等。

0-5-5 事件驱动程序设计语言

这是一种被设计成依据用户动作作响应处理的语言。它与过程化语言正好相反。过程化语言只依据菜单中所列、用户可能有的动作来响应处理。

0-5-6 工具

工具 (Tools) 是一些 Visual Basic 程序中的分类对象。这些工具是用于在窗体上产生控制工具的 (通常都是复制其本身)。

0-5-7 工程

工程 (Project) 由一群文件集合而成。这些文件有组织的产生一个工作程序。所以 Visual Basic 项目是个可执行的程序, 又称为应用程序。

0-5-8 应用程序

一个应用程序 (Application) 是一个完整的程序。

0-5-9 可执行文件

一个可执行文件 (Executable File) 又称为执行文件, 是一个已被转换成可供机器读取的指令程序。执行文件可用于其它不同的程序语言环境, 每一种程序语言都可产生执行文件, 但某一些程序语言产生的速度较快。在 Windows 环境下, 一个执行文件通常都有一个 .EXE 的扩展名。

0-5-10 自制控件

一个自制控件 (Custom Control) 是一个加强 Visual Basic 能力的程序。这些对象都不是 Visual Basic 内部的项目。通常都是由小型的软件公司开发, 且以邮购的方式销售。

若一个应用程序须有日历的功能, 不必自行开发一个日历, 可使用一个具有日历功能的自制控件来满足需求。这些对象使用起来如 Visual Basic 本身的对象一般, 但在 Visual Basic 中却未提供。必须个别另外购买。

0-5-11 图标

一个图标 (Icon) 按钮是一个小图片代表被最小化的应用程序。有时你必须将所有的应用程序最小化, 才可看见所有的图标。

0-6 工具的使用

你必须学习许多不同的技巧, 才可熟练的使用 Visual Basic 语言。第一个技巧是用户界面的设计。Microsoft Visual Basic 语言有时被称为图形用户界面 (GUI, Graphical User Interface) 的创造者。通过大量可视化工具的运用, 可使程序员设计出“所见即所得”的应用程序。

你可回想一下某些 Windows 程序，例如 CorelDRAW 及 Microsoft Write 的菜单及工具条都是这种界面。喜爱 Windows 系统的用户，对于此种界面感到非常地亲切。本书主要就是介绍如何来编写 Visual Basic 应用程序的图形用户界面。在书中将逐一地介绍各个图形用户界面工具，直至全部都明了为止。

每一个对象或是“控制项”（例如，按钮、标签、文本框）都有各自的属性集合。这些属性依照控制项的不同而有异。属性的定义与对象的特点（例如，颜色、大小、文字名称）有关。

因为对象的属性已有定义，故大部分有关属性的代码已不须编写。对 C 语言程序员而言，属性的定义是件奇怪的事，但在减少程序设计工作量这方面，却是助益良多。

一旦你了解如何建立一个用户界面，你就可学习如何写 BASIC 语言的语句，来产生一个用户界面。这些有关 Visual Basic 的按钮及控制的语句称为代码，这些代码是程序的控制项及动作的详细指令。

Visual Basic 下 BASIC 语言的运行，使得 Visual Basic 成为一个功能强大的工具。比起其它的数据库集成软件，Visual Basic 提供了更多控制你的电脑的能力。本书同时也对 BASIC 语言结构有详细的说明。

0-7 事件驱动语言的使用

Visual Basic 是一个事件驱动程序语言。也就是说，Visual Basic 程序的程序码可控制所有的用户事件及动作。当用户启动一个事件时，操作系统会告诉程序此事件为何（例如：鼠标点按一下或鼠标移动了）。

程序员可编写一些相对应的代码，使程序作出合适的响应。当响应处理结束后，Visual Basic 程序就会回复到等待的状态，并准备处理下一个事件。

0-8 启动 Visual Basic

若你还未执行安装操作的话，请先安装 Visual Basic。在安装之后，请点按“开始”菜单中的“程序”选项，再从“程序”所打开的菜单中点按“Visual Basic4.0”选项，则会呈现如图 0-1 的 Visual Basic4.0 系统菜单，从其中点选“Visual Basic4.0 中文普及版”选项来启动 Visual Basic。

如果你是在 Windows3.1 下执行 Visual Basic，则你可以从程序管理器中，点按二下 Visual Basic4.0 中文普及版图标来启动 Visual Basic。

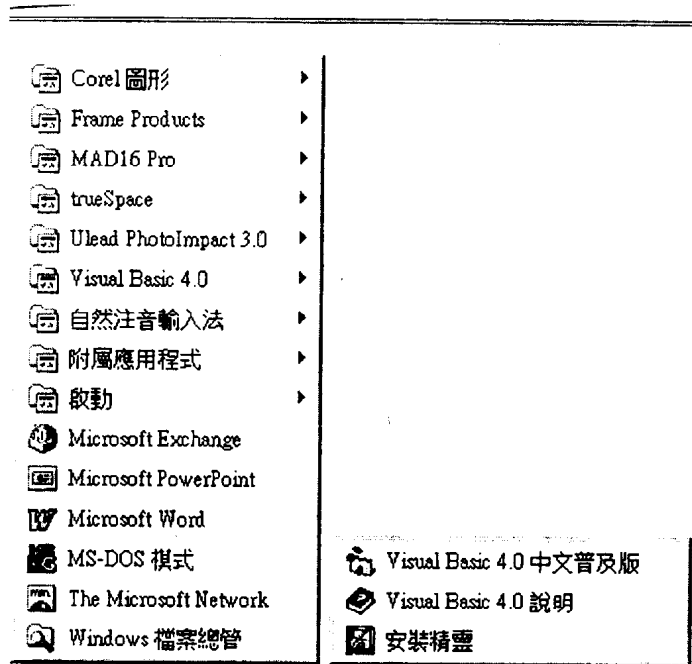


图 0-1

第 1 章

Visual Basic 透视

当你告诉程序设计员一种新语言时，他通常都会这样说“让我看一看它的程序码我就可知道它是什么了”。

以传统的 COBOL、Pascal 及 C 语言而言，程序设计员都是在这些语言内编写指令。整个程序是由这些指令编辑成的许多文件所组成的，若你想要了解整个程序是如何运作的，你只要阅读程序码及注释就能明了。

但在 Visual Basic 的程序开发中，你必须要知道编写程序码不是主要的工作，而仅是所有操作过程中的一项而已。

在 Visual Basic 中，程序设计员必须重新调整自己的观念。因为 Visual Basic 是一种包含了语言的完整程序开发系统。

在应用程序中，程序码只是其中的一部分，而其他部分就是建立对象。也就是说，整个程序结构和其它语言开发的程序是不一样的。在本章中，我们会由一些实例了解这些不同点为何。

1-1 启动 Visual Basic

在你开始构建一个 Visual Basic 应用程序前，必须打开集成开发环境 (IDE, Integrated Development Environment)。首先，请点选“开始”菜单中的“程序”选项，再从“程序”所打开的菜单中点选“Visual Basic4.0 选项，则会呈现如下的 Visual Basic4.0 系统菜单，从中点选“Visual Basic4.0 中文普及版”选项来启动 Visual Basic。如图 1-1 所示。

选择“帮助”菜单中的“内容”选项来开启如下图所示的帮助窗口。这是很重要的，因为如果你在使用 Visual Basic 时有任何疑问，你即可从帮助窗口（如图 1-2）中去找寻你要的解答。

帮助 (Help) 是提供如何使用系统的在线帮助查询工具。所有关于帮助工具的解释，请参阅第 4 章。

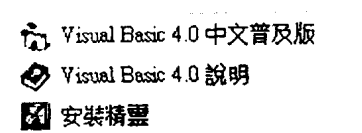


图 1-1