

SECOND EDITION

30th
ANNIVERSARY

Running
Visual Basic™
for Windows™

ROSS NELSON

*A Hands-On Introduction to
Programming for Windows*

运行 Visual Basic for Windows



学苑出版社



TP311
1251

TP311
1251-1

计算机语言技术系列丛书

运行 *Visual Basic 3.0 for Windows*

Ross Nelson 著
陆卫民 吴红艳 译
希 望 审校



一九九五年三月廿一日



30738795

学苑出版社

1994.

738795

(京)新登字 151 号

内 容 提 要

本书全面论述了 Visual Basic 语言的基本要素,详细讲述了 Visual Basic 的窗体及其设计工具、代码调试与错误处理、文件 I/O、图形程序设计、数据库、对象链接与嵌入等高级专题。本书最后以一个完整的 Windows 应用例程为例,说明了用 Visual Basic 设计 Windows 应用程序的基本步骤。

本书内容翔实,语言浅显易懂,举例丰富,是用户学习 Windows 编程的优秀教科书。

欲购本书的用户,请直接与北京 8721 信箱资料部联系,电话 2562329,邮码 100080。

版 权 声 明

本书英文版由 Microsoft 公司属下的 Microsoft 出版社(Microsoft Press)出版。版权归 Microsoft 公司所有。

本书中文版版权由 Microsoft Press 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,本书的任何部分都不得以任何形式或任何手段复制或传播。

计算机语言技术系列丛书

运行 Visual Basic 3.0 for Windows

著 者: Ross Nelson
翻 译: 陆卫民 吴红艳
审 校: 希 望
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
印 刷: 兰空印刷厂
开 本: 787×1092 1/16
印 张: 16.5 字 数: 373 千字
印 数: 1~5000 册
版 次: 1994 年 5 月北京第 1 版第 1 次
ISBN7-5077-0776-8/TP·8
本册定价: 45.00 元

学苑版图书印、装错误可随时退换

致 谢

首先,我要感谢 Visual Basic 项目组,是他们开始策划与实现了 Visual Basic,并进一步改善 Visual Basic。显然,没有这一新颖产品及其开发者的启发,写出这本书是不可能的。

其次,我还要感谢在编写本书的整个过程中给我鼓励和支持的 Claudette Moore。在 Microsoft 出版社开始写大纲时, Eric Stroo 的热情、Mike Halvorson 和 Mary DeJong 的意见给了我非常大的帮助; Mary Renaud 对书中词句的修改和润色, Dail Magee, Jr. 细心的技术编辑和错误更正工作为我避免了大量的麻烦。在 Microsoft 出版社,许多人在幕后做了大量工作,包括排字、绘图、校对、编写索引等,虽然我从来没有见过他们,但请接受我对他们所做工作的诚挚谢意。

最后,我必须感谢 Rufus 在此期间给予我的后勤支持,并特别感谢我的朋友们,尤其是 John 和 Elsa Nimmo,他们总是善意地忍受我对截止日期和编辑工作的抱怨。

目 录

引言	(1)
历史回顾	(1)
用 Visual Basic 编写 Windows 程序	(2)
第一章 Visual Basic 入门	(4)
1.1 安装和启动 Visual Basic	(4)
1.2 Visual Basic 的屏幕布局	(4)
1.2.1 主窗口	(5)
1.2.2 项目窗口	(6)
1.2.3 窗体窗口和工具箱	(7)
1.2.4 属性窗口	(8)
1.3 创建用户界面	(9)
1.3.1 设置窗体的属性	(9)
1.3.2 增加显示对象	(10)
1.3.3 完成窗体设计	(12)
1.4 编写代码	(13)
1.4.1 事件过程	(13)
1.5 运行程序	(17)
1.6 设计复审	(17)
1.7 保存程序	(19)
第二章 数据与指令	(21)
2.1 计算机与数据	(21)
2.1.1 数值表示	(22)
2.1.2 字符表示	(23)
2.1.3 位、字节与字	(23)
2.1.4 信息存储	(24)
2.2 计算机指令	(24)
2.2.1 编译程序和解释程序	(25)
2.2.2 编译 Stopwatch 程序	(27)
第三章 变量与常量	(29)
3.1 Visual Basic 的数据类型	(29)
3.1.1 数值的范围	(30)
3.1.2 浮点数	(30)
3.2 创建变量和常量	(31)
3.2.1 命名规则	(33)
3.3 创建用户定义的类型	(34)
3.3.1 自定义类型的优点	(35)
3.3.2 易混淆的术语	(35)

3.4	作用域	(36)
3.4.1	局部变量和模块级变量	(38)
3.4.2	全程变量	(38)
3.4.3	代码模块	(40)
3.4.4	缺省声明	(40)
第四章	语句、表达式与过程	(42)
4.1	赋值语句	(42)
4.2	表达式和运算符	(42)
4.2.1	表达式的类型	(43)
4.2.2	算术运算符	(43)
4.2.3	字符串拼接运算符	(47)
4.2.4	比较运算符	(48)
4.2.5	逻辑运算符	(49)
4.2.6	优先级	(51)
4.2.7	常见错误	(52)
4.3	过程	(54)
4.3.1	过程和函数的句法	(54)
4.3.2	方法	(56)
第五章	Visual Basic 的窗体与工具	(57)
5.1	Visual Basic 对象	(57)
5.1.1	本章说明	(57)
5.1.2	句法说明	(57)
5.2	窗体	(58)
5.2.1	属性	(58)
5.2.2	事件	(60)
5.2.3	过程与方法	(60)
5.2.4	应用程序 Smile	(61)
5.3	图片框和图像控件	(64)
5.3.1	属性	(64)
5.3.2	事件	(65)
5.3.3	过程和方法	(65)
5.3.4	方向板应用程序	(65)
5.4	标签	(67)
5.4.1	属性	(67)
5.4.2	事件	(67)
5.4.3	过程和方法	(67)
5.4.4	修改 Stopwatch 应用程序	(68)
5.5	正文框	(69)
5.5.1	属性	(69)

5.5.2	事 件	(70)
5.5.3	过程和方法	(70)
5.5.4	应用程序 NoteEdit	(70)
5.6	命令按钮、复选框和单选钮	(71)
5.6.1	属 性	(72)
5.6.2	事 件	(72)
5.6.3	过程和方法	(72)
5.6.4	应用程序 Game	(72)
5.7	框 架	(74)
5.7.1	属 性	(74)
5.7.2	事 件	(74)
5.7.3	过程和方法	(75)
5.7.4	修改 Game 应用程序	(75)
5.8	列表框	(75)
5.8.1	属 性	(76)
5.8.2	事 件	(76)
5.8.3	过程和方法	(77)
5.8.4	应用程序 Cheese	(77)
5.9	组合框	(78)
5.9.1	属 性	(78)
5.9.2	事 件	(79)
5.9.3	过程和方法	(79)
5.9.4	应用程序 Flight	(79)
5.10	水平滚动条和垂直滚动条	(81)
5.10.1	属 性	(81)
5.10.2	事 件	(82)
5.10.3	过程和方法	(82)
5.10.4	应用程序 Scroll Bar	(82)
5.11	计时器	(83)
5.11.1	属 性	(84)
5.11.2	事 件	(84)
5.11.3	过程和方法	(84)
5.11.4	应用程序 Metronome	(84)
5.12	线和形状	(85)
5.12.1	属 性	(86)
5.12.2	事 件	(86)
5.12.3	过程和方法	(86)
5.12.4	项目 Layer	(86)
5.13	驱动器、目录和文件列表框	(88)

5.13.1	属 性	(88)
5.13.2	事 件	(88)
5.13.3	过程和方法	(89)
5.14	其它 Visual Basic 工具	(89)
5.14.1	网格工具	(89)
5.14.2	数据	(89)
5.14.3	OLE 2.0	(89)
5.14.4	AUTOLOAD.MAK 文件	(90)
5.15	普通对话框	(90)
5.15.1	Open	(91)
5.15.2	Save As	(92)
5.15.3	Color	(92)
5.15.4	Font	(93)
5.15.5	Print	(94)
5.15.6	Help	(95)
5.16	菜 单	(96)
5.16.1	其它菜单设计选项	(100)
5.16.2	弹出式菜单	(101)
第六章	程序控制	(103)
6.1	作出选择	(103)
6.1.1	If 语句	(103)
6.1.2	多种选择	(104)
6.1.3	应用程序 Slots	(106)
6.2	循 环	(111)
6.2.1	应用程序 Banker	(113)
6.2.2	计数循环	(115)
6.2.3	是否有未想到的问题	(117)
6.3	使用数组	(117)
6.3.1	为数组申请内存空间	(118)
6.3.2	数组应用一例	(118)
6.3.3	数组与循环	(120)
6.3.4	在运行时确定数组的长度	(121)
6.4	使用控件数组	(122)
第七章	过程与函数	(124)
7.1	普通过程	(124)
7.1.1	控制流	(125)
7.1.2	参 数	(125)
7.2	函数过程	(127)
7.3	编写过程	(128)

7.3.1	进一步考虑	(131)
7.3.2	通用化	(131)
7.4	参数传递	(133)
7.4.1	传地址	(133)
7.4.2	传 值	(134)
7.4.3	传递数组参数	(135)
7.5	局部内存分配	(137)
7.6	作为参数的对象	(137)
7.6.1	应用程序 Evade	(138)
7.7	高级事件过程	(139)
7.7.1	鼠标器事件	(139)
7.7.2	键盘事件	(141)
7.7.3	拖动动作	(142)
第八章	调试与错误处理	(147)
8.1	调试工具	(147)
8.1.1	调试窗口	(148)
8.1.2	观察表达式	(149)
8.1.3	调用树	(150)
8.1.4	断点和 Stop 语句	(151)
8.1.5	观察点	(152)
8.1.6	单步调试	(152)
8.2	调试策略	(153)
8.3	请求帮助	(154)
8.4	错误处理	(156)
第九章	文件输入和输出	(160)
9.1	文件系统概述	(160)
9.2	文件操作	(160)
9.2.1	创建 TestFile 应用程序	(161)
9.2.2	执行和存储 TestFile 应用程序	(165)
9.3	顺序文件	(165)
9.3.1	打开和关闭顺序文件	(166)
9.3.2	读顺序文件	(167)
9.3.3	写顺序文件	(167)
9.3.4	Configurator 应用程序	(167)
9.3.5	忽略行边界	(170)
9.4	随机文件	(170)
9.4.1	打开和关闭随机文件	(171)
9.4.2	读写随机文件	(172)
9.4.3	Weasel 数据库应用程序	(173)

9.5 配置	(185)
9.5.1 动态连接库	(186)
9.5.2 Memory 应用程序	(188)
第十章 图形编程	(190)
10.1 窗体和图片框	(190)
10.1.1 屏幕管理	(190)
10.1.2 屏幕坐标	(192)
10.1.3 颜色	(197)
10.1.4 其它画图属性和方法	(200)
10.2 打印机对象	(204)
10.3 图形程序的硬件考虑	(205)
第十一章 数据库与网格	(206)
11.1 何谓数据库	(206)
11.2 通过 Visual Basic 存取和显示数据	(209)
11.2.1 Data 控件	(209)
11.2.2 建立联系	(209)
11.2.3 显示控件与 Data 控件的结合	(210)
11.2.4 记录集的定义	(211)
11.3 程序化控制	(211)
11.3.1 查找特殊记录	(212)
11.3.2 创建一个 Rolodex-Style 应用程序	(212)
11.4 数据库的动态访问	(214)
11.4.1 集合	(214)
11.4.2 示例表达式	(214)
11.4.3 Grid 控件	(215)
11.4.4 数据浏览应用程序	(216)
第十二章 对象的链接与嵌入	(220)
12.1 Visual Basic 中的 OLE	(220)
12.1.1 一些 OLE 术语	(220)
12.1.2 一个 OLE 例子:在程序设计时嵌入一个对象	(221)
12.1.3 链接与嵌入	(222)
12.1.4 可以链接和嵌入哪些东西	(223)
12.1.5 一个 OLE 例子:在运行时进行 OLE 控制	(223)
12.2 直接对象控制	(229)
12.2.1 对象互不相同	(229)
12.2.2 一个 OLE Automation 示例:计算器应用程序	(229)
12.3 关于 DDE	(232)
第十三章 一个完整的 Windows 应用程序	(233)
13.1 总体设计	(233)

13.2	菜单设计.....	(234)
13.3	窗体设计.....	(236)
13.4	全程声明.....	(238)
13.5	辅助窗体.....	(239)
13.6	Main 过程	(242)
13.7	事件结构.....	(243)
13.8	一点修饰.....	(250)
13.9	展望.....	(252)

引 言

对于编写 Windows 应用程序的人来说,Microsoft 公司的 Visual Basic for Windows 编程系统是一项令人兴奋的进步。Visual Basic 具有事件驱动的编程机制,新颖、易用的可视设计工具。它充分利用了 Windows 图形环境的优点,让开发人员快速地构造功能强大的应用程序。

但是,经验不足的程序员可能会猜疑 Visual Basic 是否适用。您也许已经掌握了诸如 Microsoft Excel 或 Microsoft PowerPoint 这样的应用程序,但却希望用计算机做更多的工作;或许您正在寻觅一种新的尝试,而对计算机编程发生兴趣;或许您已发现没有现成的商用程序能够完全符合自己的特定要求,所以正在考虑自己编写程序,以满足自己的特殊需要。

如果上述猜测之一成立,那么这本书正是您要找的书。本书是关于 Visual Basic 编程的初级读物。它虽然不能使您成为一个计算机专家,一个精明的玩家,或一个有希望成功的软件工程师,但它说明了怎样利用 Visual Basic,将您的想法转变为计算机的功能程序,以解决您的实际问题。

为什么会开发 Visual Basic 呢?为了了解 Visual Basic 是什么、以及为什么它在图形计算领域会产生这么大的震动,我们有必要先回顾一下历史。

历史回顾

最早的编程语言设计于 1950 年,其目的主要是为了解决复杂的数学问题。对于普通人来说,它是相当难懂的,但这在当时并不是一个严重的问题,因为当时只有在重要的研究室里才有计算机。当然,后来人们发现计算机技术的用途远不仅限于数学领域,它在企业和大学已得到了广泛的应用。当越来越多的人使用计算机时,深奥和复杂的编程语言就成了一种障碍。

因此,在六十年代初期,为适应当时的形势,Dartmouth 学院发明了一种称为“BASIC”的语言。原版本的 BASIC(Beginner's All-purpose Symbolic Instruction Code 的缩写)是一种非常简单的语言,设计它的特殊意图就是使学习编程简单化。一整代程序员在 BASIC 上付出了辛勤劳动,并用它编写了大量的程序。

BASIC 的简单性也使得它很小,当计算机也开始变小时语言的大小是至关重要的。麻省理工学院的 Altair 计算机诞生于 1975 年,它迎来了微机的革命。Microsoft 公司的共同缔造者 Bill Gates 和 Paul Allen 承担了为 Altair 开发一个 BASIC 版本的任务,它在计算机仅有的 4K 内存中运行,该 BASIC 版本最后发展成为个人计算机工业中最流行的产品。

若干年来,该编程语言得到了进一步的完善和发展。当早期的微机让位于 IBM PC 时,Microsoft 公司的 GW-BASIC 成为了标准。后来,对软件的快速、小型化和易用的要求导致了 Microsoft QuickBasic 语言的开发。QuickBasic 使 BASIC 语言加入到八十年代编程语言技术的行列,但不久,马上又出现了一种更大的变革——图形用户界面(GUI)。

Microsoft Windows 软件的出现,为 PC 用户提供了一个直观的、图形丰富的工作环境。图形用户界面使应用程序更易于学习和使用,用户只要简单地用鼠标器按钮点一下“菜单”中的选项就行了,而不必键入较长的命令。屏幕上的多窗口使用户可同时运行多个程序,当程序需要信息或需要用户决策时会出现对话框。

虽然 Windows 环境对于用户来说是相当出色的,但对于程序员来说,工作就变得比较困难了。程序员必须编写程序,去创建窗口、菜单、字体、对话框以及其它各种构件,甚至编写最简单的程序也不能例外。因此,在采用 Microsoft Windows 时,程序员是喜忧掺半——喜的是 Windows 给程序员提供了编写图形界面友好的应用程序平台,忧的是编程工作越来越复杂。

一个在屏幕上显示一条信息的简单程序在 MS-DOS 下编写时只需要 4 行语句,而在 Windows 下编写的类似程序需要 2~3 页代码;此外,程序员还要学习如何控制字体、菜单、窗口、内存和其它系统资源。但 Windows 对最终用户的好处是不容置疑的,数量众多的用户在购买 Windows 下的应用程序,专业程序员也只好痛下决心,开始编写冗长的代码。

许多人认为,Windows 预示着业余程序员的末日。在 MS-DOS 环境中,非计算机领域的专业人员一般都可学会足够的编程知识,编写简单的应用程序,以解决工作中的问题,如繁杂的计算及快速组织数据等。但在 Windows 中,最简单程序的编程工作都这么复杂,非计算机领域的专业人员还能进行编程工作吗?

用 Visual Basic 编写 Windows 程序

当 1991 年 Microsoft 公司开发出 Visual Basic 时,上述问题得到了肯定的回答。Visual Basic 编程系统用一种非常巧妙的方法将 Windows 的编程复杂性封装起来。Visual Basic 综合运用了 Basic 语言和新的可视设计工具,它既未牺牲 Windows 为之闻名的优良性能和图形工作环境,同时又提供了编程的简易性。菜单、字体、对话框、滚动正文域和所有其它构件的设计都相当容易,而且控制这些构件只需要为数不多的几行程序。

Visual Basic 也是第一批采用事件驱动编程机制的计算机语言之一。事件驱动是一种非常适用于图形用户界面的编程方式。传统的编程是一种面向过程,按顺序进行的工作,很象烹饪菜肴的指令:打鸡蛋、加牛奶、加糖搅拌、烤 20 分钟。这种编程方式的缺点是写菜谱(程序)的人总是要关心什么时候要发生什么事情。这对于烹饪来说是可以接受的,但在现代的计算机应用中,目的是让用户来操纵程序的运行。

这实际上就是事件驱动程序所要解决的问题。程序员只要编写响应用户动作的程序,如选择命令、在窗口中按鼠标器按钮、移动鼠标等,而不必编写按精确次序执行的每个步骤。我们不必编写一个大型程序,而是创建一个由若干个小程序组成的应用程序,这些小程序都由用户启动的事件来激发。利用 Visual Basic,程序员可以以空前的速度,方便地编写此类应用程序。

Visual Basic 1.0 版已获得了巨大的成功,出售了数万份拷贝并获得了大多数重要计算机杂志的表扬。1992 年秋天,Microsoft 推出了 Visual Basic 2.0 版,它提供了许多重要的新功能和特性。

在 Visual Basic 3.0 版推出后,Visual Basic 已成为一套成熟的编程系统,它具备许多功能强大的编程工具。Visual Basic 3.0 版包括以下新的特性:

- 性能的提高
- 数据库创建工具
- 使用 Data 控件进行可视数据存取,不编写代码就可以创建数据浏览应用程序
- 新 OLE(对象链接与嵌入)控件允许实地编辑对象
- 一系列普通对话框简化并统一了用户界面的设计工作
- 在应用程序中创建弹出式菜单的能力

《运行 Visual Basic 3.0 for Windows》(本书)介绍了 Visual Basic 的 3.0 版,帮助程序员提高实际编程水平。在阅读本书时,您不必知道很多的计算机专业词汇,也不必有丰富的编程经验,但在计算机上必须装有 3.0 版以上的 Windows 程序,当然,还必须熟悉 Windows 操作系统及一个以上 Windows 应用程序。如果您原来为試算表或字处理程序创建过宏程序,这将会对学习编程有帮助作用。最后,您不必害怕代数知识,这本书不包含很多的数学。但请记住,计算机语言起源于数学计算,因此,现代的计算机语言中都保留了一些数学计算功能。

学习编程实际上是学习一种新语言,确实需要一定的努力和实践。正如学习弹钢琴一样,在能弹奏交响乐之前,必须练习音阶和琶音。当读完这本书时,您还不能指望会编写与交响乐难度相当的程序。因为,本书着重的是基础知识,不会包括编程的所有内容,甚至也未完全包括 Visual Basic 的全部。但在读完本书后,您就可以建造一些有用的、不复杂的应用程序,以满足自己的需要。更重要的是可以马上掌握其它一些应用程序的编程技术,例如,Microsoft Word for Windows 用 Basic 的一个版本作为其宏语言,Microsoft Access(一种新的数据库管理程序)的编程技术基础与 Visual Basic 相同。

Visual Basic 的优良工具、能力和特性使得更为广泛的用户可以掌握编程技术,甚至掌握复杂的图形计算领域的编程技术。好啦,现在是您将计算机投入使用的时候了。

第一章 Visual Basic 入门

为了说明 Visual Basic 的易学性和用其编写实用应用程序的速度,本章将立即投入编程,并介绍一个简单的程序。实际上,使用 Visual Basic 是感性认识 Visual Basic 的最佳方法,这也将展露 Visual Basic 令人振奋的潜力。在本章的实例介绍之后,第二章和第三章将提供关于 Visual Basic 内部机制的详细说明。

1.1 安装和启动 Visual Basic

和大多数 Windows 应用程序一样,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,而实际上只要简单地执行以下四个步骤:

1. 在 MS-DOS 提示符下键入 Win 并按回车键,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 程序管理员(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. 遵循屏幕上的提示,并回答 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,并在程序管理员(Program Manager)窗口中加入一个 Visual Basic 组窗口和图标。

启动 Visual Basic 最简单的方法就是在程序管理员窗口中双击 * Visual Basic 图标,也可以双击 Windows 文件管理员窗口中代表 Visual Basic 执行文件 VB.EXE 的图标。另外,还可以通过在 MS-DOS 提示符下键入 win vb 命令同时启动 Windows 和 Visual Basic。图 1.1 说明了 Visual Basic 的三种启动方法。

1.2 Visual Basic 的屏幕布局

启动 Visual Basic 之后,屏幕上会出现五个窗口,如图 1.2 所示(在读者的 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 的空窗口;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,而属性窗口之下是项目(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。使应用程序缩小成图标可以解决屏幕混乱的问题。从程序管理员的选项(Options)菜单选择 Minimize On Use

* 双击(double click)——将鼠标指针置于某个对象之上,再连接两下鼠标左键。

选项是有帮助作用的。当选中该选项后启动新程序时,程序管理员窗口自动缩小成图标。

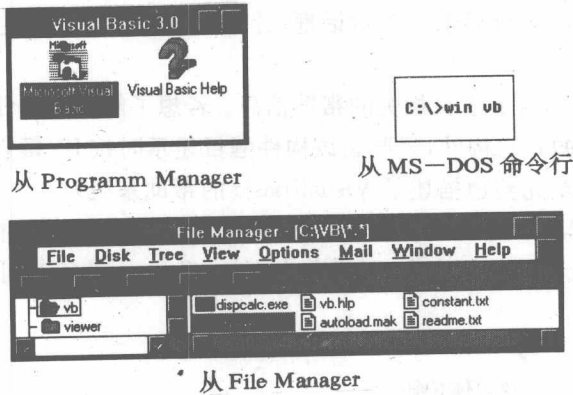


图 1.1 启动 Visual Basic 的三种方法

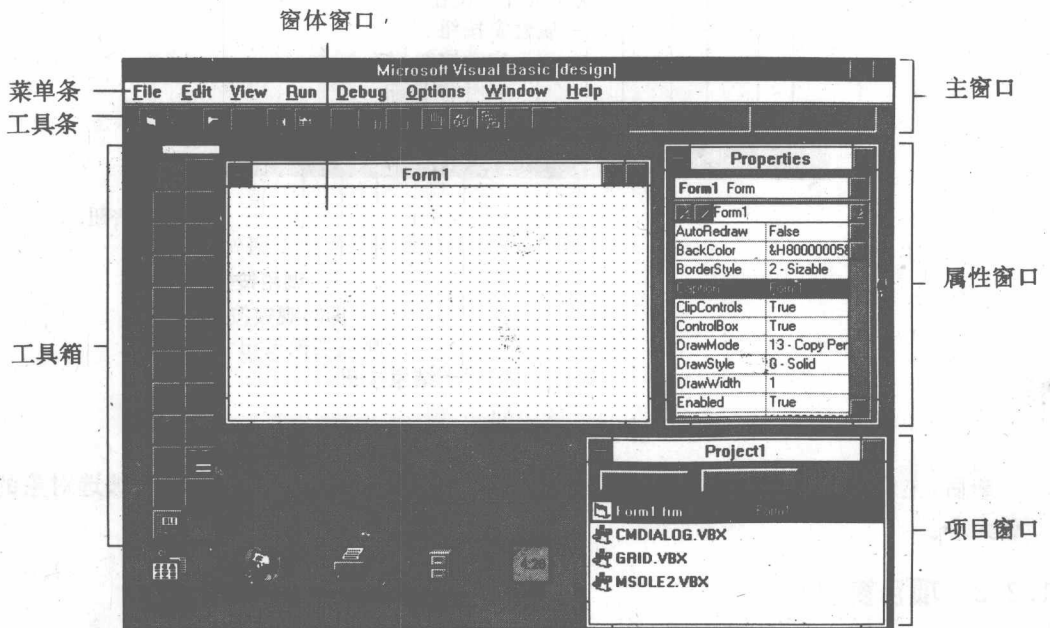


图 1.2 Visual Basic 的屏幕布局

也许您会抱怨这样太复杂了!它确实比较复杂,但降低复杂性的关键是组织,因此,Visual Basic 环境设计的目标就是保持屏幕的组织性。下面,让我们分别考察一下这五个窗口。

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 Microsoft 产品支持服务机构的联系方法。现在,我们来考察一下复杂的 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,让您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,只要在该构件醒目显示时按 F1 键,帮助系统将立即显示相关的帮助信息(第八章完整地描述了 Visual Basic 的帮助系统)。

主窗口还包含工具条。图 1.3 中工具条上的按钮是常用命令的捷径按钮。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。

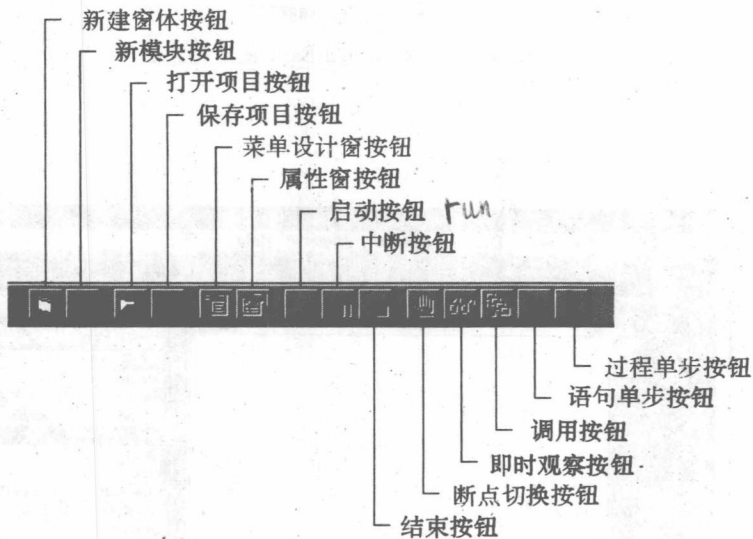


图 1.3 Visual Basic 工具条

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

1.2.2 项目窗口

项目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,项目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 的窗体窗口相关联的。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)。

* 单击(click)——将鼠标指针置于某个对象之上,再按一下鼠标左键。请与双击比较。

——译者注