SAMS

北京科海培训中心

精通 VISUAL BASIC™ 3 编程

第二版

[美] Gary Entsminger 著
冯建华 陶文中 译
赵 军 审校

清华大学出版社

SAMS 北京科海培训中心

精通 Visual Basic 3 编程

第二版

[美] Gary Entsminger 著

冯建华 陶文中 译

赵 军 审校

清华大学出版社

Secrets of the Visual Basic 3 Masters

Second Edition

图书在版编目(CIP)数据

出版者：清华大学出版社(北京清华大学校内，邮编 100084)
责任编辑：夏非彼
印刷者：门头沟胶印厂
发行：新华书店总店北京科技发行所
开本：16　印张：31.5　字数：763 千字
版次：1994 年 11 月第 1 版　　1994 年 11 月第 1 次印刷
印数：00001～10000
书号：ISBN 7-302-01696-8/TP・739
定价：64.00 元

致 谢

我衷心感谢在出版过程中，对本书进行阅读、编辑、注释或者是提供代码、软件或项目自定义控制项的人们，他们是：Sams 公司的 Greg Croy 和 Ella Davis，Microsoft 公司的 Troy Strain，Crescent Software 公司的 Don Malin，Systems 的 Jim Nelch，Ward Systems 的 Steve Ward，MicroHelp 的 Mike Novisoff，Rocky Mountain Biological Laboratory 的 Alison Brody 和 Billy Barr，Sheridan Software 公司的 Joe Modica。

作者介绍

Gary Entsminger 先生是有五年以上资历的《Micro Cornucopia》杂志的编辑。他曾在《Computer Language》，《Dr. Dobb's Journal》，《AI Expert》，《Midnight Engineering》，《Midnight Express》，《Neural Network News》，《AI Week》和《Turbo Technix》(Borland 公司的技术杂志)等杂志上发表了 100 多篇文章，并写过一本有关面向对象编程的介绍性书籍，而且是《The Tao of Object》(M&T，1990)，《The Turbo Pascal for Windows Bible》(Sams，1990)，《Secrets of the Visual Basic for Windows Masters》(Sams，1992)等书的作者。

引 言

首先 Microsoft Windows(以下简称 Windows)是大多数 PC 用户所必然选择的图形用户界面(GUI)。Microsoft Windows 的用户和程序员越来越多,用户数量的增多将增加对编写应用程序的程序员的需求。如果你是一位 PC 机上的程序员,你就必然会想到要掌握 Windows 的编程技能。

到现在为止,一个要创建 Microsoft Windows 应用程序的开发者需要学习低层的 600 个或更多的功能函数,然后用汇编语言或 C 语言来编写应用程序。从最好的方面看,这也是一项困难的、费时的任务;从最坏的方面看,这只是一个限于专业编程者的方案,甚至可以说是在虐待程序员。

在 1991 年,Borland 公司国际部发行了一个世界级版本的编译器——Turbo Pascal,也就是 Turbo Pascal for Windows,这使得 Microsoft Windows 的应用程序开发变得容易了些(它也使得 Windows 应用程序的开发不一定需要专业的程序员了)。这个编译器是我的第二本书——《Turbo Pascal for Windows Bible》(Sams,1991)所描述的对象。

在 1991 年,Microsoft 也推出了 Visual Basic,这是一个可与 C 相当的 Windows 应用程序开发工具。Visual Basic 具备了 Turbo Pascal for Windows 的许多优点,但是,与 Turbo Pascal for Windows 相比,它简化了 Windows 应用程序的开发过程。

Visual Basic 的"可视性"和"面向事件"的特征是简化 Windows 程序开发的关键。简而言之,程序员可以用图表来开发应用程序,只要点中和单击,就可从菜单中选中对象、控制项、属性等等。用 Visual Basic 来编写界面比传统的编程方法更接近用户级。当然,你必须通过点中和单击菜单项来向窗口(在 Visual Basic 中称为窗体)添加代码,好在系统已提供了模板、自动缩格、超快速语法检查等功能来使添加代码的工作简单化。

Visual Basic 完全是在 Windows 环境下的编程环境,它有几个优点:

- 因为它的内核与当前版本的 QuickBASIC 兼容(只有一些例外),所以对于当前的上百万 QuickBASIC 程序员来说只要花很少的功夫就能学会它。
- 它把 600 个 Windows 的应用程序编程接口(API)中的大多数功能函数抽取到高层,这样允许程序员直接使用 Windows 函数(按钮、对话框、菜单等)而不用 Windows 的软件开发工具箱 SDK。程序仍然可以通过一个动态链接库(DLL)来调用底层的 Windows API 功能函数。
- 程序员可以在 Windows 环境中编写、编译、运行和调试应用程序,而不必退到 DOS 下。目前尚无一个流行的其他 Windows 编译器(包括 Borland 的 C++或 Turbo Pascal for Windows)能很好地做到这一点。例如,Turbo Pascal for Windows 调试器,即使在运行一个 Windows 应用程序时,在调试过程中从图形方式的屏幕转换到文本方式的屏幕,这总是很不方便的。
- 把资源开发合并到现场,这仅在同一个 Visual Basic 环境内。你不必运行一个资源编辑器或编译器来向应用程序添加菜单、对话框、控制项等。

理论家、程序开发人员和初学者都可以用这个灵活的、面向事件(和对象)的系统来开发中级和高级的 Windows 应用程序。本书针对中级和中高级的开发者来深入研究 Visual Basic。我打算列出各种现象,利用先进的技术,让开发者掌握使用这个令人着迷的编程系统和语言来开发应用程序的技术。同时还要描述面向事件和对象的编程方式和 Windows 风格,并且通过对大量应用程序的逐步解说来使你掌握 Visual Basic 的许多高深的单元——DLL,动态数据交换(DDE)和对象链接和嵌入(OLE)的客户控制项。

如果你掌握 Basic 编程,则不用放弃 Basic 就能学会 Windows 编程,而与此同时你将拥有计算机世界中最好的两方面——结构化的 Basic 编程和访问 Windows。更重要的是,由于 Visual Basic 的通用和较快的速度(在许多测试中,它与 C++或 Turbo Pascal for Windows 的速度一样),它可以实用于商业应用程序。Visual Basic 是对 Basic 编程方法的加强,它对编程市场是一个冲击(可能是令人惊叹的),而 Basic(Visual Basic 的前身)已经落后了。

另外,在我们所测试的应用程序中,编译 Visual Basic 应用程序比用 C 和 C++编译器快许多倍,这要归功于 Visual Basic 中的解释式的语法检查器。因此,如果你要升级 Windows 程序,使它快速运行,或者要建立一个大的应用程序模型,很自然应选择 Visual Basic 系统。因为 Visual Basic 应用程序运行时和 C 或 C++应用程序一样快,或者说是在许多方面几乎一样快,所以用 Visual Basic 进行软件开发耗时很少。

Microsoft 和其他软件商提供的自定义控制项更是增强了 Visual Basic 系统,使之得到扩充。本书深入地描述了扩充 Visual Basic 的不同方法。包括自定义控制项(custom controls),DDE,DLL 和 OLE 客户控制项。

OLE 允许 Windows 程序员创建的应用程序能显示其他应用程序的数据,并能在创建某数据的应用程序中编辑该数据。换句话说,OLE 是在其他应用程序和用户的应用程序中进行 DDE(动态数据交换)的控制。

对这些高级扩充功能的使用,是要建立一个 Microsoft 称之为"基于文件"的计算机应用观点。通常,程序员和用户可建立基于应用的观点,这时一个完整的任务是单个应用程序。基于文件的观点,允许任意数量的工具作用于一个文件,这是一种非常高级的多任务方式。

例如,你的应用程序(在 Visual Basic 中建立的)可能要用到 Microsoft Excel,Quattro Pro for Windows 或其他的电子表格和数据库中来控制和处理数据;用一个图形应用程序或工具来画出数据(如 Paintbrush);用一个文本编辑器来变换数据的外貌;用 Windows 的记录器来运行一个作用于数据的宏;用你的 Visual Basic 应用程序来筛选数据和控制操作。Visual Basic 所支持的基于文件的特性,在多任务世界和复杂进程中都迈出了巨大的一步。

虽然我在书中使用的是 Visual Basic 的例子和应用程序,但更主要的是要揭示 Windows 和面向事件及对象编程的秘密。我想它对所有要掌握 Microsoft Windows 应用程序开发技术的人员(不仅是对 Visual Basic 程序员)都是很有用的。

本书列举了丰富的例子来说明现实世界的问题及其想法与编程技术之间的巧妙接口,并说明面向事件和对象的 Visual Basic 编程技术怎样使一些高级的课题既易于理解又易于管理。这些高级的课题包括:图形学、文本和对象编辑器、神经元网络、随机性和复杂性理论、图形建模、剪贴板、程序间的动态数据交换、动态链接库、基于文件的计算等等。

因为 Windows 是一个图形环境,所以必须意识到像 Visual Basic 这样的用艺术性手段来表达的语言应该是面向图形的。本书着重于应用程序的显示和图形外表,使你通过观察

(用图解来进行解释)和实际操作进行学习。

本书重点包括:

- 怎样最有利地运用 Visual Basic IDE 和 Windows
- 怎样排除错误
- 怎样用项目来建立和设计复杂的应用程序
- 怎样编写快捷、优化和有效使用内存的程序
- 怎样叙述模块范围
- 怎样组织、编写和测试大的应用程序
- 怎样扩充 Visual Basic 来使用自定义控制项,DLL,DDE,Windows API 等
- 怎样使用 OLE
- 怎样在建立大的应用程序时利用 Microsoft 和其他软件商的支持
- 怎样对待或掌握 Visual Basic 的几个弱点

在本书后半部分例子中,将讨论用 Windows API 和其他软件商的软件来扩充 Visual Basic。相信把 Windows 和 Visual Basic(包括许多高级的、复杂的工具和软件商提供的支持)结合在一起,就可作为 90 年代的商业和高校开发者开发高级的、尖端的计算机系统的工具。

在深入地了解 Windows 和 Visual Basic 的同时,你可以开发几个复杂的应用程序(一个 Wizard 项目编辑器,一个 OLE 对象编辑器和一个股票市场建模工具),你可以把它们当作单独的应用程序,或者作为创建自己的 Visual Basic 应用程序的模型或起点。

总的来说,我希望本书能在 Visual Basic 编程系统之外给予你帮助。编写本书也使我自己更好地掌握了 Visual Basic 和 Windows。

本书内容的组织

《精通 Visual Basic 3 编程》一书共 15 章,它解释和演示了 Visual Basic 的功能。特别是,本书重点在于事件驱动的编程方法,以及利用自定义控制项,DDE 和 DLL 来写出中级到高级的 Visual Basic 应用程序的方法。书中用许多例子来说明快捷和优化的技术,同时也说明了 Visual Basic 的局限。

本书的最后包括参考书目、术语和基于 Microsoft 专用工具箱及第三方软件商自定义控制项包中的自定义控制项的许多附加应用程序。因为许多销售商已经大方地向本书捐献了自定义控制项,所以你可以不再买别的软件而用这些自定义控制项来编写高级的应用程序。最后一节中的附加应用程序用自定义控制项来创建 OLE 应用程序、标准标尺、键状态指示器、类三维应用程序、使用自定义对话框、旋转控制项和瞬时滚动条的应用程序。

虽然我建议读者顺序地通读本书,但你也可以不按顺序读——各章的例子都是独立的。

本书的约定

本书无论是在列表中还是在正文中,小一号的字体用来表示代码。表示代码连续的图标——→说明 Visual Basic 中的一行在书中一行写不下,只能分开写,所以在 Visual Basic 中输入该行时,应该把由该图标连接起来的行都输入到一行中,不要分行,否则 Visual Basic 将会产生一个出错信息。

目 录

第 2 部分 Visual Basic 3.0 的开发

第1部分 Visual Basic 3.0的发展与概述

第1章 Visual Basic系统

- 面向对象程序设计简历
- 面向对象的程序设计，Visual Basic 风格
- 面向事件的程序设计工作的实质
- 运用 Change 事件来触发自动的和实时的更新
- 运用 Focus 事件来自动初始化动作

1.1 对象、事件和动态程序设计

当开发工具和程序设计环境改变时，对开发应用程序方法的选择毫无疑问也要受到影响。我认为在20世纪80年代的中期和后期，大多数程序员考虑得更多的是程序设计中的算法而不是界面。换句话说，程序员的精力集中于完成任务的过程，而不是用户与该过程之间的交互方式。如果有时间，某个程序员可能编些界面，但这不是必需的。因此，当时运行许多程序的过程只是启动、进行某项工作，然后退出或中止。有时，当程序员认为用户界面是次要的或是无关紧要的部分时，甚至连最基本的菜单都不创建。

在 Windows 中开发应用程序时，必须意识到界面不能是次要的了，这种意识需要有稍为更广阔的程序设计视野。即要看清楚应用程序在整个系统(或主机或环境)中的位置，以及它与系统的最大联系(这里是 Windows)及与系统中其他元素(Windows 中的其他应用程序)的联系。

不在 Windows 这样的系统中工作的程序员常常很容易就会编写出不去考虑其他应用程序的程序。快速的、清晰的编程无疑是许多程序员努力要达到的理想境界。但是，快速、不清晰地程序设计也常常被认为是足够好的，因为在 DOS 下要实现应用程序在高层次上的互相联系是不容易的。所以，许多程序员都会忽视其他的应用程序，在编写代码时认为只有操作系统而没有其他应用程序，这种情况不包括中断和驻留(TSR)应用程序。

Windows 为不同形式的高层次交互都提供了相应的机制：应用程序之间、操作系统和应用程序之间、公共的共享代码库和数据库之间。但是，程序员要从这些机制中得到好处，必须完全地接受 Windows 系统，也就是说必须给予界面足够的重视。

某个应用程序可能没有菜单项、控制项等等，但必须指定某种形式的窗口来包含该应用程序，还要使应用程序对从 Windows 上接收到的消息(messages)作出反应。在本书中你将会得到有关消息的详细说明，现在只要把它们想象为鼠标单击、鼠标双击、对计时事件的按键和从端口或其他应用程序来的输入等之类的事情。应用程序所在的窗口就是应用程序与 Windows 环境相联系的地方，Windows 通过该窗口向应用程序的过程传送所有必要的信息。

在Visual Basic中,应用程序窗口被称为窗体(forms)。窗体中可包括控制项(controls),控制项也可以与Windows或其他的窗体和控制项交互作用,甚至可以与其他的应用程序交互作用。在Visual Basic中所有的项(窗体、控制项、自定义控制项、打印机等)都被称为对象(objects)。有时这些对象能漂亮地集中在一起,有时却不能。怎样扩展Visual Basic,从而处理这些对象的相互关系是本书的重点。

例如,一个应用程序的窗体对怎样更多和更好地与Windows相联系上就有很大的灵活性,它可以使用菜单项、命令按钮、DDE链接等。本章先介绍Windows怎样传送信息,然后介绍一个好的Windows应用程序怎样与Windows相连接并交互。Visual Basic中的对象和事件程序设计机制可以帮助读者快速地开发出Windows上的复杂应用程序。本章将展示一个宏伟的蓝图,即描绘Windows和Visual Basic一起工作的方法。

1.2 应用程序开发初步

你若曾经使用过一种程序设计语言,毫无疑问,你对怎样开发一个应用程序会有自己的想法,无论是在Windows上或其他的操作系统上。要开发一个Windows应用程序,应掌握:

- Windows是怎样与一个应用程序交换信息的;怎样准备你的应用程序,使它能在一个Windows的窗口里运行。
- 怎样用对象和事件方法来开发应用程序。这两个要求看起来很粗略,但开发一个Windows应用程序的主要思想仍然是:要习惯于写出代码来解决问题——写出用于描述和处理信息(或数据)的代码。

开发DOS应用程序和Windows应用程序的主要区别是:在DOS上必须自己开发用户界面(除非是编写一个非常基本的或不需要界面的应用程序)。如果编写一个Windows应用程序,必须使该程序具有与Windows GUI相连接的某种界面。但是,要使应用程序只具有一个简单的界面是容易的,因为不必一点一点地从琐碎的地方开始建立,只要利用Windows的界面,然后继续使用你所掌握的大多数程序设计方法:变量,过程,函数,返回值,if-then-else,循环,数组,程序结构等。

要开发一个能解决某个问题的Windows应用程序,必须:

1. 与Windows进行交互联系(用Visual Basic设计中的窗体和控制项对象)。
2. 取得输入应用程序的信息(输入单元)。
3. 保存信息(数据单元)。
4. 用过程、函数或对象方法来处理信息(操作单元)。
5. 取得结果(输出单元)。

这几步很简单,可能除了与Windows的连接外,其他内容你都很熟悉。要与Windows连接,只要用Visual Basic控制项和窗体即可。

取得、保存和处理信息及获得结果(第2到第5步)都是用传统的结构化编程方法来处理应用程序的。可以使用条件语句(if-then)和循环(while,repeat)等,可以把操作按段或组划分,并命名(过程,函数,对象),还可以把任务关联或对象关联的过程、函数和定义包含在模块中。需要再次说明的是这种程序设计过程与你已经掌握的过程可能一样(无论是在Basic或其他任何一种结构化语言,如C或Pascal中进行程序设计)。

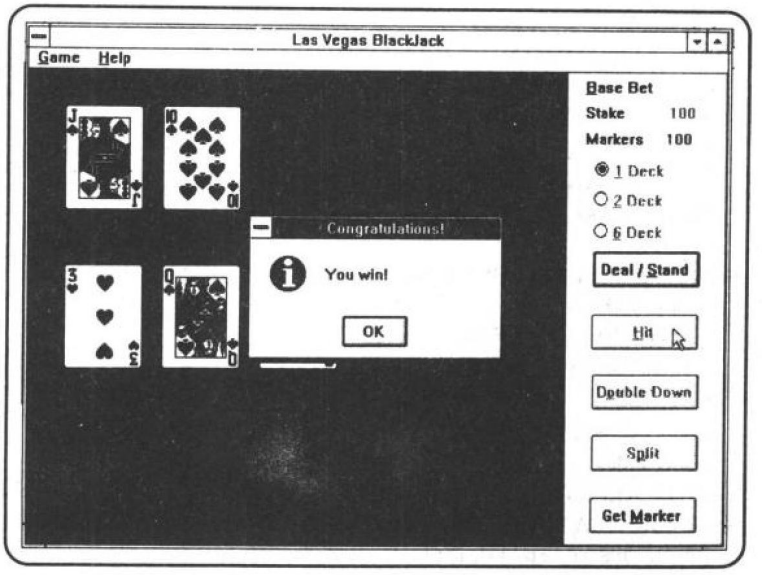

图 B.4　BlackJack：游戏取胜

```
        AnalyzeHand
      Else
        If DoubleButton.Enabled = False Then
            DoubleButton.Enabled = True
        End If
        If HitButton.Enabled = False Then
            HitButton.Enabled = True
        End If
        ' If the player has only two cards, and the stake
        ' is greater than or equal to two bets
        If HowManyCards(ActivePlayer) = 2 And
            Stake& >= (Bet(PLAYER) + Bet(PLAYERB)) Then
            If DoubleButton.Enabled = False Then
                DoubleButton.Enabled = True
            End If
        End If
      End If
    Else
      AnalyzeHand
    End If
Else
  ' He wants a deal — check bet amount
  If Bet(PLAYER) <= 0 Then
    MsgBox "You must bet something or I can't deal!", 64, "Bad Bet"
  ElseIf Bet(PLAYER) > Stake& Then
    MsgBox "You can't bet what you don't have!" + CRLF$ +
    "Hint — Get a marker.", 64, "Dealer Cannot Extend Credit!"
  ElseIf Stake& <= 0 Then
    MsgBox "You don't have a stake." +
    CRLF$ + "Please get a marker at the cage.", 64, "You Are Broke"
  Else ' Bet amount is OK. Go ahead and deal
    Deal
```