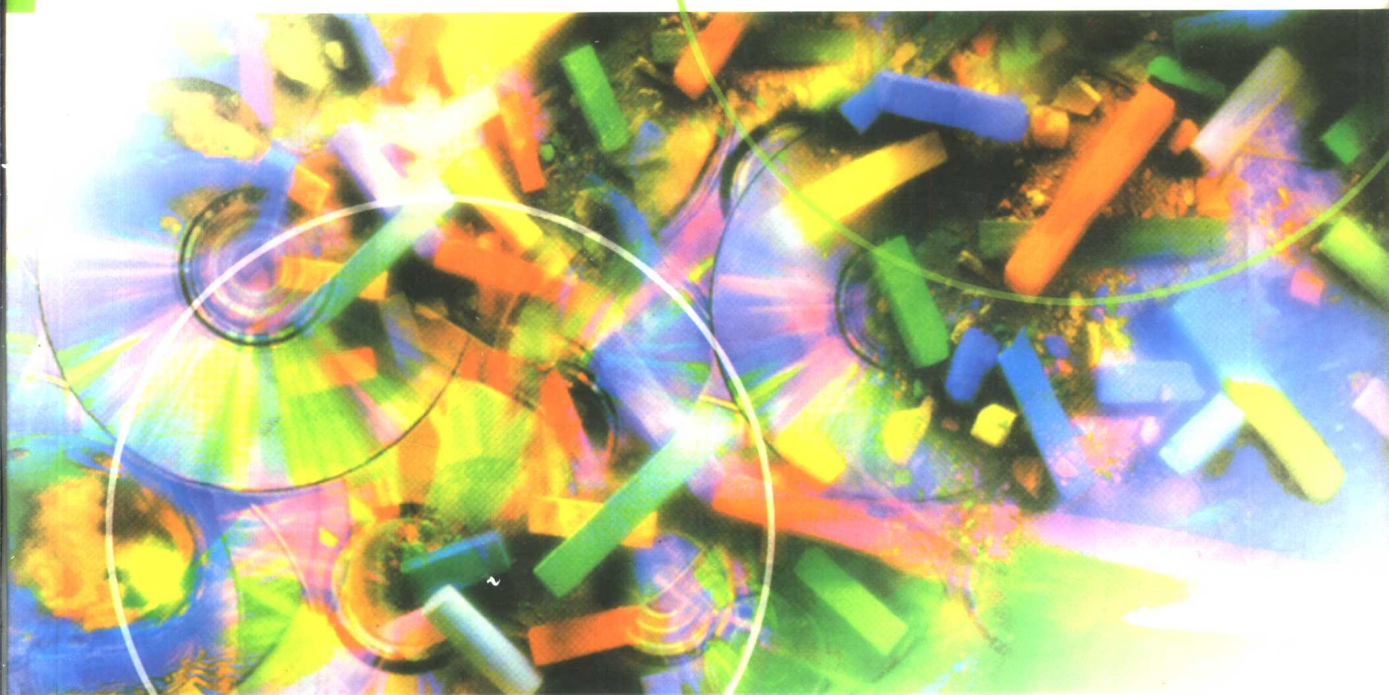


Peter
Norton

〔美〕 Peter Norton 编著
陈庆章 伊俊敏 译

教你学

Visual
Basic 6



que®



浙江科学技术出版社

Peter Norton 教你学 Visual Basic 6

[美] Peter Norton 编著
陈庆章 伊俊敏等 译

浙江科学技术出版社

Peter Norton: Peter Norton's Guide to Visual Basic 6 from the English
language edition published by Que® Corporation

Copyright© 1999 by Que® Corporation

All Rights Reserved. For Sale in Mainland China Only

本书中文简体字版由浙江科学技术出版社和 Que® 公司合作出版，
未经出版者书面许可，本书任何部分均不得以任何方式复制或抄袭。
本书封底贴有防伪标签，无标签者不得销售。

版权所有，翻印必究。

浙江省版权局著作权合同登记号 图字：11-1998-68 号

书 名	Peter Norton 教你学 Visual Basic 6
编 著	[美] Peter Norton
译 者	陈庆章 伊俊敏等
出 版	浙江科学技术出版社
印 刷	杭州富春印务有限公司
发 行	浙江省新华书店
制 作	浙江科学技术出版社计算机图书工作室
读者热线	0571-5157523
电子信箱	Hzzjkj@mail. hz. zj. cn
开 本	787×1092 1/16
印 张	33.5
字 数	810 000
版 次	1999 年 7 月第一版
印 次	1999 年 7 月第一次印刷
书 号	ISBN 7-5341-1297-4/TP·101
定 价	47.50 元
责任编辑	熊盛新
封面设计	孙 菁

作 者 序

我经常想是什么使一些人冒那么大的险来从事滑翔和跳伞这样的危险运动呢？显然从飞机上或山崖上跳下来是不全得到钱的，甚至还会伤着的。有时候看起来，从崖顶或飞机上跳下来并相信几磅重的尼龙绳就能使你安全落地，简直是一件花钱却无益的事。我知道从飞机上跳下来是很过瘾的，但如果降落伞不能张开还谈得上过瘾吗？

同样的道理，许多人也能体会到使用像 Visual Basic 之类的工具开发 Windows 应用程序的酸甜苦辣。总的来说，编写计算机程序是一项艰苦的事，对大多数人来说都不简单。我确信许多跳伞爱好者会说，一个人没日没夜地从在计算机屏幕前不断地搞那些程序代码肯定是什么地方不正常。

老实说，花几个小时用 Visual Basic 或其他开发系统编写一个可以工作的 Windows 应用程序的确不是一件容易的事情。像我这样的人，是以掌握编程语言并使计算机作出以前从未作过的工作为乐趣的。而且用编程语言完成一些大作还能获得不菲的钞票。

无疑爱好跳伞和开发 Windows 应用程序还是有共同之处的，两者要掌握相关的设备和工具。跳伞爱好者不学习使用高程计就会有麻烦的，同样 Visual Basic 开发人员不学习使用调试工具，碰到错误时就会一筹莫展。

用 Visual Basic 编程与跳伞最大的不同是，用 Visual Basic 创建 Windows 应用程序是不可能伤着自己的。当然你要追求目标，但编程过程是不会摔断你的骨头的。

一、本书的读者对象

本书是为某些特定读者所写的。虽然书中的很多材料对有经验的 Visual Basic 开发人员也是有趣的，但本书的主要读者对象是那些新用 Visual Basic 但对应用程序开发却不陌生的人。因此读本书之前不需要任何使用 Visual Basic 的经历，但以前的编程语言经历对学习本书是很有用的。Visual Basic 吸引了各种各样的开发人员。Visual Basic 的开发人员包括无任何应用程序开发经历的新人，以及有多年 COBOL 或 RPG 应用程序开发经验的大型机开发老手。作为新入 Visual Basic 道的程序员，你很可能想知道 Visual Basic 究竟是什么，用 Visual Basic 能开发什么程序，以及开始用这一复杂的开发环境之前要掌握些什么。

二、为什么学习 Visual Basic

10 年前我还是一个 C 语言程序员，编写在 UNIX 下运行的应用程序。C 语言是一种功能十分强大的开发系统。实际上没有任何程序用 C 或 C++(面向对象的 C 语言)干不了的。但无疑，要成为 C/C++ 的开发高手，必须很下功夫。C 和 C++ 这样功能强大的语言也充满了各种加密算法、方法和复杂的语法。要熟练运用 C/C++ 语言，必须掌握几百个重要的 C/C++

函数和编程约定。掌握 C++ 不仅需要科学的方法，还需要执着的追求，必须花大量的精力来学习它。

我们可能都听说过 BASIC 语言由来的故事。在 20 世纪 60 年代，Dartmouth 学院的讲师 John Kemeny 需要一种能让学生接受的计算机编程语言。因为当时的编程语言极其难用，属于专业级的工具，对初学者来说很困难。Kemeny 决定建立一套全新的语言，它既易于学习，又能提供当时复杂计算机语言的许多功能。他最后创造出 BASIC——初学者通用符号指令码。BASIC 成功了，因为它既简单但又功能强大，只需要很少的命令就可以执行传统语言需要上千条语句才能完成的任务。

从那时到现在过去几十年了，老的 BASIC 语言已进化为 Visual Basic 了，这是一种健壮的面向对象的语言，非常适合编写 Windows 应用程序。Visual Basic 语法是 Microsoft Office 97 等内置脚本语言的核心。Word、Excel、Access 和 PowerPoint 全都共用 Visual Basic for Application(简称 VBA)，它的语法与 Visual Basic 6 的是一模一样的。

1996 年，Microsoft 开始向第三方软件商发放 VBA 使用许可。这就是说 Microsoft Office 家族之外的应用程序也可以共享与 Word、Excel、Access 和 PowerPoint 一样的编程语言。例如 Autodesk 公司的计算机辅助设计应用程序，AutoCAD 和 Visio 公司的面向对象的图像软件——Visio，都是第一批采用 VBA 的第三方产品。

除了 Office 97 支持全部 VBA 语法之外，像 Outlook 和 Internet Explorer 之类的应用程序都支持 VBScript。VBScript 是 VBA 语言的一个简化版本，非常适合不需和不能用完整 VBA 的场合。如果 IE 内有全部 VBA 语法，就可以编写出“特洛伊木马”式的应用程序，用户只要查看远程 HTTP 服务器上的一个 Web 页，硬盘数据就可能被毁，后果不堪设想。但即使去掉了某些语言元素，VBScript 还是一种用于 Web 页和 Microsoft Outlook 的有用脚本语言。只要熟悉 Visual Basic，用 VBScript 编程就没有任何问题。

这就是说 Visual Basic 技巧还可用于许多其他系统的编程。不必再学习编程语言，就能编程与 Word、Excel 和 Visio 等程序集成应用。

熟练的 Visual Basic 开发人员也是熟练的 Word、Excel、Access 和 PowerPoint 开发人员。开发在 Word 或 Excel 中的开发与 Visual Basic 中开发的唯一区别是如何获取应用程序后的代码。编写自定义 Word 应用程序(如图像或文档管理工具)与使用 Visual Basic 有同样的编码约定。

这种能将编程技巧移植到 Microsoft Office 之类重要和成功产品的能力，足以值得我们学习掌握 Visual Basic 这种开发环境。

BASIC 语言的一大优点是许多命令本身就是英语日常用的动词和名词。例如，要获取(get)文件中的数据，我们用 Get 命令；使一个隐藏的对象重现在屏幕上，我们将其 Visible 属性设为 True，等等。如果不能确定要用的具体语法，不妨用英语直觉，按已掌握的 Visual Basic 知识试试最合乎逻辑的形式。

Visual Basic 继承了 BASIC 的简单、功能强大和易学的特点，但为满足广大开发人员的需要，功能进行全面的扩充。例如 Visual Basic 4 开始增加了向 Visual Basic 应用程序添加自己的对象类型的功能。对象是一种像顾客或雇员一样的物理实体的模型。对象包含描述对象的属性，告诉对象发生什么的事件和提供对象支持的操作的方法。对象及其属性与方法将在本书中深入讨论，详见第二部分“面向对象的编程技术”。

三、Visual Basic 是正确选择吗？

选择 Visual Basic 作为主要开发平台有错吗？现在有许多其他成功的开发环境可用，如 Visual C++、Borland Delphi 和 PowerSoft 的 PowerBuilder。无疑选择错误系统的可能性是很大的，开发工具商可能失败，可能不断改进并支持其产品。

几年以来桌面计算环境与今天的大不一样。1990 年，大多数台式机运行于 DOS 下，流行的开发系统是 FoxPro、Clipper 和 dBASE。严格的编程语言仅有汇编、C 和 Pascal。Microsoft 的 Windows 还未流行也不成功。当时许多开发人员在获取平台技巧上花了不少功夫和精力，但这些平台却很快无用了，因为它们是基于 DOS 的，无法移植到 Windows 中。还有的是因为无法满足用户不断增长的需要。可以说许多开发系统的失败是因为它们没有跟上桌面计算快速的发展变化。

但对 Visual Basic 来说，情况却不同，Visual Basic 的前途光明是基于以下事实：首先 Visual Basic 是一个宏大的产品。数以百万计的不同水平的开发人员都在用 Visual Basic 编写应用程序。Visual Basic 现在已进行了 6 次大的改动，并且还继续受 Microsoft 的发展和改进。许多先进技术，在其他 Microsoft 产品中出现后不久就引入到了 Visual Basic 中。

而且，Visual Basic 是一种功能强大的开发系统。VBA 语言本身带有内置的函数和执行常见任务的子过程。除了其独有特性外，Visual Basic 还提供了创建自定义库和对象的功能，这些库和对象可以在运行时装入或者与可发布的应用程序捆绑一起。本书相当一部分内容就是讲述如何建立这些除内置功能外的自定义功能的。

Visual Basic 也被第三方产品广泛支持。面向应用程序开发人员的计算机杂志上经常刊登 Visual Basic 外接程序和开发工具的广告。几十种杂志和快报经常刊登有关 Visual Basic 主题和技术的最新文章。讲述 Visual Basic 的专门书籍也有几十本。

最后，Visual Basic 是 Microsoft 的产品，Microsoft 无疑是世界上最大也最成功的软件公司。Microsoft 对 Visual Basic 和 VBA 的投入在过去几年里也增长迅速，并且推出面向不同水平开发人员的版本。Visual Basic 6 就有 3 个版本：标准版、专业版和企业版。每一种版本都包括专门面向该层次开发人员的特性，并且价格也与类似产品有竞争性。基于以上这些原因，Visual Basic 无疑是当前最具吸引力的开发系统。与过去被淘汰的系统相比，Visual Basic 的前途无比光明。你也可以沾沾 Visual Basic 成功的光。

四、本书的组织方式

本文的设计遵循典型的编程项目，首先本书介绍 Visual Basic 环境让读者知道开发 Visual Basic 应用程序是怎么回事。然后本书阐述如何创建全功能的、专业级应用程序，让开发人员环境之外的人也能使用。这就是说应用程序必须包括错误处理和复杂的用户界面设计。

本书分为 7 个部分，各部分的安排尽量按照循序渐进的原则进行。这 7 个部分如下：

第一部分“使用 Visual Basic 6”有 9 章，是想让读者熟悉 Visual Basic 编程环境，并为创建复杂应用程序作好准备。具体章节包括 Visual Basic 6 环境介绍、处理控件和窗体，以

及 Visual Basic 应用程序的代码简介。接下来是使用 Visual Basic 变量和建立自定义工具栏来增强用户界面等复杂内容。

第二部分“面向对象的编程技术”专门讲述使用 Visual Basic 6 进行面向对象的编程。虽然读者可能会想为什么这一高级论题那么早就开始讲述，这是因为 Visual Basic 6 特别依赖于对象和对象类，所以尽早接触这一主题。本部分的 3 章细述了创建对象类的过程，不仅包括 Visual Basic 内置的对象类，也包括为应用程序创建的自定义对象类。最后是创建 ActiveX 控件的过程和 Visual Basic 应用程序的自动化服务器。读者很快就会发现 Visual Basic 开发时使用类和自定义 ActiveX 控件的好处。

第三部分“创建打印输出效果”讲述如何为 Visual Basic 应用程序建立打印输出。虽然 Crystal Reports for Visual Basic 这个外接程序是建立报表和其他打印输出的常用方法，但并不仅限于此。实际上，不依靠 Visual Basic 控件或其他外接程序就可以直接控制本机打印机或网络打印机。

第四部分“数据访问”共 5 章，论述 Visual Basic 的超级数据管理功能。大多数 Visual Basic 应用程序都涉及相当的数据操作。Microsoft 估计用 Visual Basic 编制的 60% 以上的应用程序的主要目的是管理数据。因此，Visual Basic 中包括强大的数据访问和操作功能就是顺理成章的了。本部分几章详细介绍了 Visual Basic 6 可用的所有数据库开发工具，实际上它讲到了用 Visual Basic 建立和高效使用数据库的所有内容。

第五部分“与其他应用程序的集成”阐述了在 Visual Basic 应用程序中使用“自动化”技术的秘密和要求。Visual Basic 有与其他 Windows 应用程序共享数据和资源的功能。通过“自动化”技术，Visual Basic 可以与控件和其他 Windows 应用程序通信。例如，可以 Microsoft Word 而不是 Crystal Reports 来为 Visual Basic 程序准备打印输出功能。或者用 Visual Basic 编写财务应用程序时可利用 Excel 的许多功能。

第六部分“在 Internet 上开发”讨论了如何使用 Visual Basic 来创建提供与 Internet 和内部网用户实时的数据库连接。Internet 现在已经是一种为可抗拒的力量。Visual Basic 是用得最广的 Web 应用程序开发平台之一，这主要是缘于它的内置 Web 开发功能。ActiveX Server Pages 与 Microsoft 的 Internet Information Server 和其他与 Internet 有关的产品一起工作，提供了 Internet 上实时访问数据库的功能。

第七部分“高级论题”讨论了给程序代码添加专业级和高级功能和如何向用户发布你的 Visual Basic 应用程序。这里假设你用 Visual Basic 6 编写的应用程序要被没有 Visual Basic 环境的其他用户使用，你应当提供专业级的修饰和包装，这样才会对你努力工作的结果感到欣慰。

本书还包括两个附录。附录 A 是词汇表，附录 B 是完整的 Visual Basic 用于窗体、模块、变量和其他组件的命名约定——《Reddick 命名约定》，这是 Visual Basic 程序用得最广泛的命名约定。

五、结束语

Visual Basic 也许是现在最激动人心的开发平台。这不仅是因为 Visual Basic 的强大功能

和易于使用，也在于学习和使用这种其他开发平台所没有的通用编程语言所带来的额外好处。以前从没有一种开发系统能提供如此多的好处。实际上任何类型的应用程序都能由 Visual Basic 编写，而且你掌握 Visual Basic 后可以写出以前从不敢想的应用程序。

笔者尽最大努力在本书中为读者提供有用的学习工具和材料。我知道如果你认为它们没有帮助，也不会对这些内容感兴趣的并买此书。相反，本书的每一章都有掌握该章主题所需的内容，这恰恰是开发人员真正要知道的东西。

如果读者对本书有什么问题和评论，欢迎与笔者联系，我的电子邮件地址为：
mgroh@anstion360.com 或者访问我的主页 *http://www.praxis.net/mgroh*，查看对本书示例代码和感兴趣的附属文件与代码的更新情况。

译者序

诺顿这个名字我们一定不会陌生。彼特·诺顿这位带着黑边眼镜、温文尔雅先生以他的智慧和勤奋为我们奉献了像 Norton Utilities 等优秀的软件。在编写程序的同时，诺顿还致力于著书立说，将自己使用计算机和编程的心得教给更多的人。

本书是诺顿与一位编程里手 Michael Groh 合作编写的，适用于新近进入 Visual Basic 编程的读者，尤其是那些有一些其他语言编程基础，想用 Visual Basic 编程的程序员们。

Visual Basic 自推出以来，一直受到大家的欢迎，现在仅在美国就有几百万的 Visual Basic 程序员。全球的数量就更不用说了，而且还有越来越多的程序员投入到 Visual Basic 编程之中。本书循序渐进，在介绍了 Visual Basic 的基本用法之后，就分专题讲述其强大的功能和实现方法，如在 Visual Basic 中用多种方法操作数据库、用 Visual Basic 开发 Internet 应用程序、与 Microsoft Office 的集成编程等实用而热门的专题。

与诺顿细致的风格一致，本书用大量的实例、严密平和的语言一步一步地指导我们进入 Visual Basic 编程的大世界，掌握 Visual Basic 各项功能的应用，学习如何设计和编写优秀的 Visual Basic 应用程序。

我们有幸能将这本书翻译出来献给大家。几个月的艰辛工作是与廖辉传(第一、三部分)、熊小芸(第一部分)、丁振凡(第二部分)、杜曼玲(第四部分)、陈庆章(第五、六)和武和雷(第七部分)的努力分不开的。全书最后由陈庆章、伊俊敏统稿校译，并完成图形制作。由于我们水平有限加之时间仓促，书中错误和疏漏之处在所难免，请广大读者不吝指正。批评和建议请发至 macroyee@163.net。

最后，希望愿投入 Visual Basic 编程的广大读者能从本书中有所收获，更好地掌握 Visual Basic 6 先进的编程技术，为我们创造出更多更好的应用程序。

目 录

第 1 章 开始使用 Visual Basic 6	1
1.1 Visual Basic: Windows 开发的快车道	1
1.1.1 为开发人员改进了的技术	2
1.1.2 理解 Visual Basic 的优点	3
1.2 获取帮助	4
1.3 Visual Basic 开发概述	5
1.3.1 设计和构建用户界面	6
1.3.2 编写响应事件的代码	7
1.3.3 按需创建和调用其他过程	9
1.3.4 测试和调试	9
1.3.5 转换为运行时版本	10
1.3.6 准备可发布的文件集合	11
1.4 首次创建 Visual Basic 应用程序	11
1.4.1 一个简单的工程	12
1.4.2 创建工程	12
1.4.3 给窗体添加控件	13
1.4.4 设置控件的属性	14
1.4.5 编写事件驱动代码	14
1.4.6 测试和调试示例应用程序	16
1.4.7 将示例应用程序转化为运行版本	16
1.5 本章小结	17
第 2 章 理解 Visual Basic 工程	19
2.1 使用不同类型的工程	19
2.1.1 选择一种工程	20
2.1.2 理解工程的不同组件	21
2.2 理解 Project Explorer 窗口	22
2.3 处理多个工程	23
2.4 设置工程的选项	23
2.4.1 使用 General 页	23
2.4.2 使用 Make 页	24
2.4.3 使用 Compile 页	24
2.5 本章小结	25
第 3 章 Visual Basic 代码基本知识	26
3.1 理解 Visual Basic 过程	26
3.1.1 理解子过程	26
3.1.2 理解函数	27
3.1.3 理解注释	28

3.2 使用 Visual Basic 编辑器	29
3.2.1 打开代码窗口	31
3.2.2 理解代码窗口的各种特性	32
3.2.3 在编辑器窗口中使用键盘和鼠标快捷键	34
3.2.4 理解编辑器窗口的各种选项	38
3.3 控制程序流程	40
3.3.1 使用条件分支结构	41
3.3.2 循 环	45
3.3.3 使用编译器指令	49
3.4 本章小结	51
第 4 章 使用 Visual Basic 变量	52
4.1 变量使用示例	52
4.2 理解 Visual Basic 数据类型	54
4.2.1 字节型	54
4.2.2 整 型	55
4.2.3 长整型	56
4.2.4 单精度型	56
4.2.5 双精度型	56
4.2.6 货币型	56
4.2.7 布尔型	56
4.2.8 日期型	57
4.2.9 字符串型	58
4.2.10 对象型	58
4.2.11 变体型	60
4.3 声明变量	60
4.3.1 隐式声明和显式声明的比较	61
4.3.2 为变量命名	62
4.3.3 强制显式声明	62
4.4 使用命名约定	63
4.5 理解变量的作用域和生命期	63
4.5.1 检查作用域	64
4.5.2 决定变量的生命期	65
4.6 数据类型转换	66
4.7 处理数组	68
4.7.1 声明数组	68
4.7.2 数组的限制性	68
4.8 理解可变数据类型	69
4.8.1 变体型数据的存储需求	70
4.8.2 变体型变量的几种特殊值	71
4.9 将变量传递给过程	73
4.9.1 可选的参数	74

4.9.2 使用命名参数	75
4.10 使用常量	75
4.11 用户自定义数据类型	76
4.12 本章小结	77
第 5 章 设计用户界面	78
5.1 理解 Visual Basic 窗体	78
5.2 套用 Windows 的外观特征	78
5.3 创建窗体	79
5.4 添加新窗体	80
5.5 理解窗体属性	81
5.6 窗体外观属性	82
5.6.1 Borderstyle 属性	82
5.6.2 Name 属性	83
5.6.3 Caption 属性	83
5.6.4 Icon 属性	83
5.6.5 Picture 属性	84
5.6.6 Backcolor 属性	84
5.6.7 与字体有关的几个属性	85
5.7 窗体控制属性	85
5.7.1 Controlbox 属性	85
5.7.2 Minbutton 和 Maxbutton 属性	86
5.8 窗体大小和位置属性	86
5.8.1 Startupposition 属性	86
5.8.2 Left、Top、Height 和 Width 属性	86
5.8.3 Movable 属性	87
5.8.4 Windowstate 属性	87
5.8.5 Showintaskbar 属性	87
5.8.6 其他属性	87
5.9 其他窗体属性	88
5.9.1 Tag 属性	88
5.9.2 Scalemode、Scaletop、Scaleleft、Scalewidth 和 Scaleheight 属性	88
5.10 引用窗体属性	89
5.11 在代码中设置属性	89
5.12 用不同方法来定位窗体	90
5.12.1 Move 方法	90
5.12.2 Zorder 方法	90
5.13 使用窗体的各种绘图方法	91
5.13.1 Circle 方法	91
5.13.2 Line 方法	91
5.13.3 Pset 方法	92
5.13.4 Point 方法	92

5.13.5	Cls 方法	93
5.13.6	Refresh 方法	93
5.13.7	Textwidth 和 Textheight 方法	93
5.13.8	Setfocus 方法	93
5.14	显示和隐藏窗体	94
5.15	装载窗体	95
5.16	卸载窗体	95
5.17	理解窗体事件	96
5.18	窗体维护事件	97
5.18.1	Initialize 事件	97
5.18.2	Load 事件	97
5.18.3	Activate 和 Deactivate 事件	97
5.18.4	Queryunload 和 Unload 事件	97
5.18.5	Terminate 事件	97
5.19	窗体操作事件	97
5.19.1	Gotfocus 和 Lostfocus 事件	98
5.19.2	Paint 事件	98
5.19.3	Resize 事件	98
5.20	窗体鼠标事件	98
5.20.1	Click 和 Dblclick 事件	98
5.20.2	Mousedown 和 Mouseup 事件	99
5.20.3	Mousemove 事件	100
5.21	窗体键盘事件	100
5.21.1	Keydown 和 Keyup 事件	100
5.21.2	KeyPress 事件	101
5.22	充分利用内置对话框	101
5.22.1	消息框	102
5.22.2	使用 Inputbox 函数	104
5.23	理解 MDI 应用程序	105
5.23.1	创建 MDI 父窗体	106
5.23.2	将普通窗体转换为 MDI 子窗体	106
5.23.3	MDI 子窗体的行为	106
5.23.4	将 MDI 父窗体作为起始窗体	106
5.23.5	创建窗体的多个实例	107
5.24	窗体练习	107
5.25	本章小结	108
第 6 章	在窗体内处理控件	109
6.1	熟悉工具箱	109
6.2	熟悉 Visual Basic 工具	110
6.3	定制工具箱	111
6.4	给窗体添加控件	111

6.5 使用网格对齐功能	112
6.6 选定多个控件	112
6.7 窗体布局功能	112
6.8 控件的属性	113
6.8.1 显示简单文本	113
6.8.2 显示一个按钮	114
6.8.3 显示列表	115
6.8.4 显示下拉列表框	116
6.8.5 显示复选框	117
6.8.6 提供互斥的选项	117
6.8.7 使用框架分组控件	118
6.8.8 输入文本	118
6.8.9 画图和显示图片	118
6.9 选择文件、颜色、字体、打印机和帮助	119
6.10 生成受时间控制的事件	120
6.10.1 OLE 容器控件	121
6.10.2 网格和电子表格控件	121
6.10.3 创建多标签式的显示	122
6.11 其他控件	123
6.12 给控件命名	124
6.13 复制控件	124
6.14 控件数组	125
6.15 引用控件	126
6.16 将控件作为参数传递	127
6.17 确定控件类型	127
6.18 使用控件集合	128
6.19 控件的焦点	128
6.20 Tab 次序设定	129
6.21 绑定控件	129
6.22 拖放操作	129
6.23 处理键盘事件	130
6.24 处理鼠标事件	131
6.25 本章小结	131
第 7 章 掌握菜单和工具栏的使用	132
7.1 理解菜单	132
7.2 使用菜单编辑器	132
7.3 菜单数组	134
7.4 菜单使用约定	134
7.4.1 使用省略号	135
7.4.2 允许键盘导航	135
7.4.3 使用快捷键	135

7.5	使用弹出式菜单	136
7.6	在运行时配置菜单	137
7.7	理解工具栏	137
7.8	查找想用的图标	137
7.9	给你的应用程序添加工具栏	138
7.10	定制工具栏	139
7.10.1	删除按钮	139
7.10.2	改变工具提示	139
7.10.3	改变图标	139
7.10.4	添加按钮	140
7.10.5	移动按钮	140
7.10.6	创建切换式(toggle-style)的按钮	141
7.10.7	创建选项式的按钮	141
7.10.8	给按钮添加标题文字	141
7.11	在运行时配置工具栏	141
7.12	处理工具栏事件	141
7.13	运行菜单和工具栏示例程序	142
7.14	本章小结	142
第 8 章	使用 Visual Basic 调试工具	143
8.1	查找错误	143
8.1.1	鉴别语法错误	144
8.1.2	鉴别逻辑错误	145
8.1.3	处理运行时的错误	146
8.1.4	合理避错	146
8.2	使用模式选项	147
8.3	传统调试技巧	148
8.3.1	使用 MsgBox 语句	148
8.3.2	使用 Debug.Print 语句	150
8.4	使用 Visual Basic 调试工具	151
8.4.1	使用调试工具栏	152
8.4.2	使用断点中断执行	153
8.4.3	逐语句执行	154
8.4.4	控制程序流程	156
8.4.5	熟悉调试窗口	157
8.4.6	使用调用堆栈	161
8.5	本章小结	163
第 9 章	处理运行时错误	164
9.1	辨别运行时错误	164
9.2	在 Visual Basic 应用程序中捕获错误	165
9.2.1	捕获错误	165
9.2.2	处理错误	167

9.2.3 重定向程序流程	167
9.3 使用 Err 对象	167
9.3.1 使用 Err.Description 属性	167
9.3.2 使用 Err.Number 属性	167
9.3.3 显示更多的可帮助信息	168
9.4 捕获何种错误	170
9.5 使用 Resume 语句	170
9.5.1 Resume Label	171
9.5.2 Resume Next	171
9.5.3 Resume 语句	172
9.6 使用 On Error Goto 0 语句	172
9.7 在过程范围内处理错误	173
9.8 记录错误	174
9.9 本章小结	175
第 10 章 创建对象和类	176
10.1 理解面向对象技术开发的好处	176
10.2 理解组件对象模型	177
10.3 从对象开始	177
10.3.1 检查对象	178
10.3.2 在应用程序中使用对象	178
10.4 理解类	179
10.5 创建一个简单的类	180
10.5.1 开始建立类模块	180
10.5.2 给类添加属性	181
10.5.3 为类创建方法	182
10.5.4 使用类模块	183
10.5.5 使用属性过程	185
10.5.6 理解属性过程类型	186
10.5.7 安全的属性过程	188
10.6 本章小结	188
第 11 章 高级类概念	189
11.1 理解初始化和终结	189
11.1.1 查看 Class_Initialize	190
11.1.2 使用 Class_Terminate	190
11.1.3 测试已有类对象	191
11.2 创建特殊类型的属性	192
11.2.1 创建只读属性	192
11.2.2 设置只写属性	193
11.2.3 设置默认的属性或方法	193
11.3 增强简单类的功能	194
11.3.1 查看 Class_Initialize 过程	194

11.3.2	添加简单的属性过程	195
11.3.3	添加 Color 属性过程	195
11.3.4	添加 Year 属性过程	196
11.3.5	给 Clscar2 添加简单方法	197
11.3.6	增强 Decelerate 方法	197
11.3.7	增强 Accelerate 方法	197
11.4	使用增强的对象类	198
11.4.1	用数据填充窗体	199
11.4.2	创建和破坏 Car 对象	199
11.4.3	设置 Car 对象的颜色	200
11.4.4	使 Clean 复选框成为只读型	200
11.5	使用对象浏览器	201
11.6	理解集合的概念	202
11.7	本章小结	202
第 12 章	处理对象和集合	203
12.1	使用集合	203
12.2	检查内置的 Visual Basic 集合	204
12.2.1	查看 Forms 集合	204
12.2.2	深入 Forms 集合	204
12.2.3	处理 Forms 集合	205
12.2.4	理解控件 Controls 集合	207
12.2.5	查看其他内置的集合	209
12.3	使用集合对象	210
12.3.1	创建集合	211
12.3.2	给集合添加对象	211
12.3.3	统计集合中项目的个数	212
12.3.4	从集合检索对象	213
12.3.5	从集合中删除项目	214
12.3.6	破坏集合	216
12.4	优化对象引用	216
12.5	建立自定义集合案例	217
12.5.1	理解 Wayward 对象	217
12.5.2	功能封装	218
12.6	创建自定义集合	219
12.6.1	从集合类开始	219
12.6.2	添加 Item 方法	220
12.6.3	对 Add 方法编程	220
12.6.4	实现 Remove 方法	221
12.6.5	实现 Paint 方法	222
12.6.6	对 Bluecount 属性编程	222
12.6.7	完成集合类	223