

秦 戈 冉小兵 刘 勇 编著



Visual BASIC 6.0

(中文版)

编程指导 与技巧指点



电子科技大学出版社

Visual BASIC 6.0

清华大学出版社

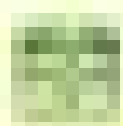


Visual BASIC 6.0

(中文版)

编程指南

与技巧指南



清华大学出版社

Visual BASIC 6.0 (中文版)

编程指导与技巧指点

秦 戈 冉小兵 刘 勇 编著

电子科技大学出版社

图书在版编目(CIP)数据

Visual BASIC 6.0 (中文版)编程指导与技巧指点/

秦戈, 冉小兵, 刘勇编著. —成都: 电子科技大学出版社, 2000.4

ISBN 7-81065-404-7/TP · 271

I. V... II. ①秦...②冉...③刘... III. BASIC 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2000)第 20975 号

内 容 提 要

Visual Basic 以其易学易用、高效率开发、功能强大等特点而受到众多编程爱好者的青睐, 而其最新版的 Visual Basic 6.0 更是已经成为一种专业化的开发语言和编程环境。本书通过大量实例, 分门别类地讨论了 Visual Basic 6.0 中各种令人激动的新特性, 以及各种实用编程技巧, 内容涉及 Visual Basic 6.0 的编程基础知识、设计程序菜单、程序调试方法、创建和使用可重用代码部件、Client/Server 编程访问数据库等。

本书内容翔实、概念清晰, 语言通俗易懂, 其中的许多编程实例可以直接在实际的应用程序中使用。另外, 在介绍程序设计的同时, 本书着重突出了设计方法的合理性在程序设计中的重要性, 并进行了详尽的剖析。

本书适用于广大 Visual Basic 爱好者、初、中级程序员、科研人员以及大中专院校师生。



声 明

本书无四川省版权防盗标识不得销售; 版权所有, 违者必究, 举报有奖。

举报电话: (028)6636481 6241146

3201496

Visual BASIC 6.0 (中文版)
编程指导与技巧指点

秦 戈 冉小兵 刘 勇 编著

出 版: 电子科技大学出版社 (成都建设北路二段四号, 邮编: 610054)

责任编辑: 文 利

发 行: 新华书店

印 刷: 四川导向印务有限公司

开 本: 787×1092 1/16 印张 20.125

字数 490 千字

版 次: 2000 年 5 月第一版

印 次: 2000 年 5 月第一次印刷

书 号: ISBN 7-81065-404-7/TP · 271

印 数: 1—3000 册

定 价: 25.80 元

前言

计算机的发展无论是从硬件还是软件，这些年来都堪称是突飞猛进。尤其是从软件技术的变化来看，随着 Microsoft 公司关于 Windows 应用程序的一系列标准的公布，程序开发的方法已经发生了本质性的改变。传统的编程理念已不再适用于现今程序的设计。我们周围一些从事程序设计开发的朋友、同事都曾提到过：对于广大编程爱好者来说，必须一开始就将程序设计方法定位在面向对象（OOP）的事件驱动编程上，因为这就是 Windows 程序的基本工作方式，必须深刻地理解和领会它，才能开发出好的 Windows 应用程序。新概念、新方法的引入固然不易，但这是技术发展的趋势，不紧跟必遭淘汰。本书正是适于这个趋势，为推进这些新方法和新技术的普及而编写的。

本书是针对 Visual Basic 6.0 中文版而编写的。经过多年的发展，Visual Basic 已经成为一种真正专业化的开发环境。使用 Visual Basic 6.0 不仅可以编写出企业级水平的客户/服务器及强大的数据库应用程序，还可以进行 Web 编程，使用浏览器，创建 Active X 部件，并能与许多其它应用程序紧密集成。本书全面介绍了这些高级的 Visual Basic 编程技术，全书共分五部分：第一部分（1~4 章）是 Visual Basic 基础知识；第二部分（5~6 章）介绍了如何调试程序和设计程序菜单；第三部分（7~8 章）详细介绍了使用对象编程和有关 DLL 及 Windows API 的知识；第四部分（9~13 章）介绍了如何创建数据库应用程序和 Web 数据库开发；最后一部分（14 章）介绍了如何为应用程序添加帮助文件。

当然，Visual Basic 最大的特点也许是它将编写功能强大的 Windows 应用程序这类繁重的工作变得简单而又十分有趣。使读者能够灵活使用 Visual Basic 中的各种编程技术，并能设计出具有一定实用价值的 Windows 应用程序是编写本书的宗旨。我们希望广大读者在通过本书的学习后，不仅能掌握利用 Visual Basic 6.0 开发 Windows 应用程序的方法，而且能对整个 Windows 编程有个深入的了解，并能举一反三，融会贯通。

由于编者水平有限，成书时间比较仓促，书中难免有不妥之处，欢迎读者提出宝贵意见。

编者

2000 年 3 月

目 录

第一章 Windows 程序设计入门	1
1.1 程序设计方法	2
1.2 Windows 的工作方式	3
1.3 Windows 面向对象的事件驱动程序设计	4
1.4 应用程序的界面设计	6
第二章 Visual Basic 的安装和启动	8
2.1 什么是 Visual Basic	9
2.1.1 Visual Basic 的由来	9
2.1.2 Visual Basic 的特点	9
2.1.3 Visual Basic 版本简介	10
2.2 Visual Basic 6.0 的新内容	11
2.2.1 数据访问的新内容	11
2.2.2 Internet 特性的新内容	13
2.2.3 部件创建的新内容	13
2.2.4 语言特性的新内容	14
2.2.5 向导的新内容	15
2.2.6 升级使用 Windows 通用控件的应用程序	16
2.3 Visual Basic 的安装	17
2.3.1 Visual Basic 安装的系统需求	17
2.3.2 Visual Basic 的安装过程	18
2.3.3 添加或删除 Visual Basic 部件	20
2.4 Visual Basic 6.0 的启动及界面简介	21
2.4.1 Visual Basic 6.0 的启动	21
2.4.2 Visual Basic 6.0 的界面简介	21
2.4.3 设置 Visual Basic 的集成开发环境	24
2.5 可视化编程的几个基本概念	25
2.5.1 对象、属性、方法和事件	25
2.5.2 窗体、控件和过程	26
2.6 建立第一个 Visual Basic 程序	27
2.6.1 用 Visual Basic 编程的一般步骤	27
2.6.2 建立第一个 Visual Basic 程序	27

第三章 Visual Basic 编程基础	31
3.1 Visual Basic 应用程序的结构	32
3.2 Visual Basic 的代码编写机制	33
3.2.1 代码模块	33
3.2.2 使用“代码编辑器”	33
3.2.3 自动完成编码	34
3.2.4 编码基础	35
3.2.5 Visual Basic 的命名约定	37
3.3 变量、常量及数据类型	38
3.3.1 变量	38
3.3.2 理解变量的范围	40
3.3.3 在编程中使用变量	41
3.3.4 静态变量	44
3.3.5 常数	45
3.3.6 数据类型	47
第四章 扩展程序的功能	54
4.1 数组的建立和引用	55
4.1.1 固定数组	55
4.1.2 动态数组	58
4.2 利用过程提高编程效率	59
4.2.1 子过程	60
4.2.2 函数过程	62
4.2.3 使用过程	63
4.2.4 向过程传递参数	64
4.3 控制程序的流程	69
4.3.1 Visual Basic 中的运算次序	69
4.3.2 选择结构的基本语句——IF 语句	70
4.3.3 多分支选择结构语句——Select Case 语句	72
4.3.4 条件循环语句——Do...Loop 语句	73
4.3.5 固定次数循环语句——For...Next 语句	75
4.3.6 针对集合元素的循环——For Each...Next 语句	76
4.4 使用控制结构	76
4.4.1 控制结构的嵌套	77
4.4.2 控制结构的退出	77
4.4.3 退出子过程或函数过程	78

第五章 程序查错与调试	80
5.1 程序中的几种错误	81
5.2 避免编译错误	82
5.3 如何处理运行错误和逻辑错误	83
5.4 调试工具如何工作	85
5.5 应用程序的三种状态	86
5.6 如何监测变量的值	87
5.6.1 使用“监视”窗口	87
5.6.2 使用“本地”窗口	89
5.6.3 使用“快速监视”	90
5.7 使用“立即”窗口	91
5.7.1 利用“立即”窗口测试表达式的值	91
5.7.2 通过“立即”窗口给变量或属性赋值	92
5.8 调用堆栈	92
5.9 程序中的断点设置	93
5.9.1 在程序中设置断点	93
5.9.2 使用 STOP 语句	94
5.10 跟踪代码的执行	95
第六章 为应用程序建立菜单	97
6.1 组织菜单项	98
6.2 建立主菜单	99
6.2.1 创建菜单控件	99
6.2.2 分隔菜单项	101
6.2.3 设置访问键和快捷键	102
6.3 菜单的命名规则	103
6.3.1 对菜单标题的赋值	103
6.3.2 确定菜单的 Name 属性	104
6.4 设计子菜单	104
6.5 为菜单控件编写代码	105
6.6 创建弹出式菜单	105
第七章 使用对象编程	109
7.1 使用对象	110
7.1.1 对象是什么	110
7.1.2 用对象能做什么	111
7.1.3 使用对象初步	112
7.1.4 对象之间的关系	114

7.1.5	创建对象	117
7.2	Visual Basic 中的对象	124
7.2.1	与对象有关的术语	124
7.2.2	找出对象所属的类	124
7.2.3	使用字符串名调用属性和方法	125
7.2.4	对一个对象执行多个动作	126
7.2.5	使用缺省属性	126
7.2.6	创建对象数组	128
7.2.7	创建对象集合	129
7.2.8	Visual Basic 集合对象	130
7.2.9	Visual Basic 中的集合	135
7.3	查找对象信息	137
7.3.1	对象浏览器的内容	138
7.3.2	查找和浏览对象	140
7.3.3	添加对象的描述	140
7.3.4	在过程之间移动	141
7.3.5	浏览其它应用程序的对象	142
7.4	创建自己的类	142
7.4.1	类是什么	142
7.4.2	定制窗体类	144
7.4.3	在程序中使用类	146
7.4.4	调试类模块	150
7.4.5	Visual Basic 窗体的存活期	151
7.4.6	类模块和标准模块的比较	156
7.5	向类中添加属性和方法	158
7.5.1	向类中添加属性	158
7.5.2	属性过程与公共变量的比较	160
7.5.3	运用属性过程	160
7.5.4	向类中添加方法	166
7.5.5	选择属性还是方法	167
7.5.6	设置缺省的属性或方法	168
7.5.7	友元属性和方法	169
7.6	向类中添加事件	171
7.6.1	声明和引发事件	171
7.6.2	处理对象的事件	173
7.6.3	WithEvents 与窗体上的控件事件的比较	176
7.6.4	向窗体添加事件	177
7.6.5	对事件的总结	178
7.7	用自己的对象编程	179

7.7.1	对象引用和引用计数.....	179
7.7.2	对象模型	180
7.8	创建自己的集合类.....	184
7.8.1	公有集合示例.....	185
7.8.2	私有集合示例.....	189
7.8.3	创建自己的集合类.....	192
7.8.4	面向对象程序设计的优点.....	195
第八章	访问 DLL 和 Windows API.....	197
8.1	在应用程序中使用 DLL 过程.....	198
8.2	访问 Microsoft Windows API	199
8.2.1	使用 API Viewer.....	199
8.2.2	将文本文件转换为 Jet 数据库文件	200
8.3	声明 DLL 过程.....	201
8.3.1	指定库	202
8.3.2	处理使用字符串的 Windows API 过程	203
8.3.3	值传递和地址引用传递.....	204
8.3.4	处理不标准的 DLL 过程名	205
8.3.5	使用序号标识 DLL 过程.....	205
8.3.6	灵活的参数类型.....	206
8.4	将字符串传递到 DLL 过程.....	206
8.4.1	将字符串传递到使用自动化的 DLL	207
8.4.2	修改字符串参数的过程.....	207
8.5	将数组传递到 DLL 过程中	208
8.6	将自定义的数据类型传递到 DLL 过程	210
8.7	将函数指针传递到 DLL 和类型库	212
8.7.1	有关函数指针的知识.....	212
8.7.2	使用 AddressOf 关键字	212
8.7.3	在变量中存储函数指针.....	213
8.7.4	子类派生	213
8.7.5	使用函数指针的局限与风险.....	215
8.8	将其它类型的信息传递到 DLL 过程	216
8.8.1	传递 Null 指针	216
8.8.2	传递属性	216
8.8.3	将句柄用于 DLL.....	217
8.8.4	传递变体值.....	217
8.9	将 C 语言声明转换为 Visual Basic 声明	218

第九章 使用 Visual Basic 访问数据	219
9.1 理解数据库	220
9.2 Visual Basic 6.0 中的全新数据访问技术	221
9.3 窗体和数据识别控件	223
9.4 Visual Basic 中的 ADO、DAO 和 RDO	225
9.5 Visual Basic 的数据源	226
9.5.1 Visual Basic 6.0 中新的数据源	226
9.5.2 使用数据识别的类和用户控件创建数据源	226
9.5.3 数据环境设计器	226
9.5.4 可创建的 ADO 记录集	226
9.6 Visual Basic 和 Microsoft Transaction Server	227
9.7 Visual Basic 中的数据绑定	227
9.8 Visual Basic 的远程特性	228
9.9 DHTML 和 Visual Basic 数据访问	230
第十章 Visual Basic 中的数据访问工具	231
10.1 “数据视图”窗口和可视化数据库工具	232
10.2 格式化对象	232
10.2.1 格式化数据的处理步骤	232
10.2.2 理解对象的分层结构	233
10.2.3 格式化对象实例	234
10.3 使用 SQL 编辑器	236
10.3.1 SQL 编辑器中的存储过程	236
10.3.2 SQL 编辑器中的触发器	238
10.4 使用 T-SQL 调试器	240
10.4.1 安装和兼容性	240
10.4.2 使用 T-SQL 调试器	240
10.4.3 疑难解答	243
10.5 用户连接设计器	243
10.5.1 插入一个新的 UserConnection 对象	244
10.5.2 插入一个新的 Query 对象	247
第十一章 使用数据报表设计器	250
11.1 数据报表设计器	251
11.2 创建一个简单数据报表	252
11.3 扩展数据报表的功能	258
11.4 向数据报表添加计算字段	260
11.5 向数据报表添加一个 Function 控件	262

11.6	数据报表中的分组信息.....	266
11.7	向数据报表添加一个合计字段.....	268
11.8	在数据报表中强制分页.....	270
11.9	向数据报表添加日期、时间、页号和标题.....	270
11.10	导出数据报表.....	272
11.10.1	ExportFormat 对象.....	272
11.10.2	显示对话框.....	273
11.10.3	ExportFormat 提供对话信息.....	273
11.10.4	模板代码.....	274
11.11	打印数据报表.....	275
11.12	Data Report 事件.....	276
11.12.1	ExportReport 和 PrintReport: 查询、同步和异步处理.....	276
11.12.2	ProcessTimeOut 事件——对于同步函数.....	277
11.12.3	Error 事件——对于异步函数.....	277
11.13	理解数据报表设计器中控件的位置.....	278
11.13.1	数据报表设计器中的控件位置.....	279
11.13.2	Command 对象的例外: 分组和总计合计.....	280
11.13.3	检查层次结构.....	280

第十二章 访问 Microsoft Jet/Microsoft Access 数据库中的数据281

12.1	创建与 Microsoft Access 数据库的连接.....	282
12.1.1	添加数据环境设计器到工程中.....	282
12.1.2	设置 Connection 属性.....	282
12.2	创建数据环境 Command 对象.....	283
12.2.1	打开数据环境设计器.....	283
12.2.2	创建 Command 对象.....	283
12.3	创建简单的数据绑定窗体.....	284
12.3.1	将 Command 对象从数据环境设计器拖放到空窗体中.....	284
12.3.2	创建浏览记录所用的命令按钮.....	285
12.4	创建基于查询的数据网格窗体.....	285
12.4.1	创建基于查询的数据环境 Command 对象.....	285
12.4.2	将 Command 对象从数据环境拖动到空窗体中.....	287
12.5	创建主表/细表窗体.....	287
12.5.1	创建一个父/子数据环境 Command 对象.....	287
12.5.2	将 Command 对象从数据环境拖动到空窗体中.....	288
12.6	创建数据绑定报表.....	289
12.6.1	创建数据环境 Command.....	289
12.6.2	打开数据报表设计器.....	289
12.6.3	将 Command 从数据环境拖动到数据报表中.....	289

12.6.4 设置报表的数据属性.....	290
第十三章 创建与 SQL Server 数据交互的 DHTML 应用程序.....	291
13.1 向 SQL Server 数据库添加表.....	292
13.1.1 打开 DHTML Application DLL 工程.....	292
13.1.2 创建与 SQL Server 数据库的连接.....	292
13.1.3 向数据库中添加表.....	293
13.2 创建数据输入 HTML 页面.....	293
13.2.1 向 HTML 页面添加标签和 TextField 元素.....	294
13.2.2 创建数据环境 Command 对象.....	294
13.2.3 添加 Button 元素向表中追加记录.....	294
13.2.4 加载 HTML 页面.....	296
13.3 通过 HTML 页面查看和更新数据.....	296
13.3.1 向新的 HTML 页面添加标签和 TextField 元素.....	296
13.3.2 修改数据环境 Command 对象以允许数据更新.....	297
13.3.3 添加代码以显示 HTML 页面上的记录.....	297
13.3.4 添加 Button 元素漫游记录.....	298
13.3.5 加载 HTML 页面.....	298
13.4 通过 HTML 页面运行存储过程.....	298
13.4.1 向数据库中添加存储过程.....	299
13.4.2 创建基于存储过程的数据环境 Command 对象.....	299
13.4.3 向新的 HTML 页面添加标签和 TextField 元素.....	299
13.4.4 向 HTML 页面添加 Select 元素.....	300
13.4.5 添加代码运行存储过程.....	300
13.4.6 加载 HTML 页面.....	302
第十四章 为应用程序添加帮助.....	303
14.1 添加对帮助的支持.....	304
14.1.1 HelpFile 属性.....	304
14.1.2 HelpContextID 属性.....	304
14.2 为窗体添加“这是什么”帮助.....	305
14.3 与应用程序一同发布帮助.....	306
14.3.1 发布 WinHelp.....	306
14.3.2 发布 HTML 帮助.....	307
14.4 WinHelp (.hlp) 文件的制作.....	307
14.4.1 编写 Help 文件的源文件——rtf 文件.....	307
14.4.2 定义 Winhelp 文件的项目文件——hpj 文件.....	309
14.4.3 与应用程序的结合.....	310

第一章

Windows 程序设计入门

本章要点

- 程序设计方法
- Windows 的工作方式
- Windows 程序设计的新特性
- 程序的界面设计

Microsoft 公司于 2000 年 3 月推出了其最新的操作系统——Windows 2000 中文版，而且它仍以最快的速度陆续推出了 Windows 2000 的日文版、俄文版等不同语言的版本。

作为一种全新的计算机操作系统，Windows 自从 90 年代初一诞生，就迅速地风靡了全球。与以往我们所使用的操作系统——DOS 不同，Windows 用若干组布置美观、意思明确的图形窗口取代了 DOS 中许多繁复难懂的命令，可以通过鼠标点取非常直观地完成许多功能，这使得计算机使用起来更容易，也使计算机的使用者们也更容易明白计算机正在干什么。正是由于这些优点，Windows 现在已经成为 PC 机上首选的操作系统。而许多软件供应商们也正在把他们以前基于 DOS 平台编写的软件改写为支持 Windows 操作系统的新版本，以适应当今电脑技术的发展。可以说，支持 Windows 操作系统的编程已经成为当今 PC 机编程的主流。

虽然基于 Windows 平台的编程对用户十分友好，但对程序开发者来说却较难编写，直到引入了像 Visual Basic 这样专业的可视化环境开发工具后，才使创建 Windows 应用程序较为容易。

在深入了解 Visual Basic 之前，让我们先来了解一下有关计算机和编程的一些历史，以及一些可适用于任何编程语言的有关编程的基本概念。理解这些有关编程基础的概念，有助于我们今后编制出更好更有效的程序。

1.1 程序设计方法

当我们真正面临一个问题，要利用计算机编程来解决它时，是不是一开始就编写程序代码呢？不！答案是否定的。前面我们已经提到过，在拿到一个需要求解的问题后，应先经思考，确定解决问题的方法，然后将其分解成一系列可实施的算法，最后用程序代码来描述你所设计的算法。说得详细一些，一个完整的程序设计过程应该包括以下的几个步骤：

问题定义 → 问题解析 → 算法设计 → 界面设计 → 代码编写 → 程序调试

所谓“问题定义”是指在程序设计的一开始，我们首先应该仔细考虑要解决的问题，能够清楚地定义问题的实质。如果不能给出问题的清楚定义，就无法很好地理解问题，也就不可能有目的地去设计程序而解决问题。因而，在接受任务时，一定要先写出有关给定任务的描述。

接下来我们需要做的是问题解析。在充分理解问题的基础上，有次序地设计解决问题的方法，写出程序要完成的任务提纲，描述解决问题的基本步骤。

然后，将所设计的解决问题的各个基本步骤用流程图或框图的形式表示出来，这一步称作构造算法（算法设计）。我们注意到，Visual Basic 对于程序执行过程中每一个相对独立的处理过程均采用“输入 → 处理 → 输出”这样一个步骤循环。有时，程序中的某些处理过程需要用户额外地给出一些新的输入，或者要求用户作出一些选择，这样，就对程序的界面设计有了更高的要求。不仅只是在程序的开始和末尾有专门的界面设计，而且在整个程序中的任何地方都应该根据程序设计的需要，随时调整程序的界面，以使我们设计的应用程序能够真正符合交互式图形界面的要求。所以，我们在进行算法设计的同时，也

应根据程序的需要，全面考虑整个程序的界面设计，并用适当的方式描绘出程序的输入和输出屏幕。

以上这些工作做完之后，就可以进行程序代码的编写了。将我们已构造好的算法和整个程序输入输出界面用编程语言代码一步步地写下来。记住，一旦开始编写代码，就应该把注意力集中在怎样用代码来描述你的程序设计上，而不要再去反复考虑你的算法是否正确，以及界面是否合适，这样有利于提高编程的效率，而且不容易出错。另外，在程序代码编写时，应该注意以下两点：

- 首先应使代码尽可能高效，为用户提供好的响应时间；
- 其次，应尽量使代码易于维护。这样才可能使将来对代码的修改以及对程序的升级变得较为容易。

要做到使代码易于维护，应保持各个代码模块的任务尽可能单一，彼此之间相对独立，使整个程序结构化；另外，在程序中适当地编写必要的注释行对程序的功能进行描述也能使程序变得更加易读。

所有代码编写完成后，应检查一下整个程序中是否有明显的错误和失误，尤其是程序中的逻辑错误。这类错误几乎无法被代码编译器所发现，但他们却会导致程序错误的输出，甚至是使程序陷入死循环而崩溃。因而，我们要求，在将程序输入计算机之前，应该仔细检查程序的各个模块之间，以及模块内每个控制流程的分支之间是否有这种逻辑混乱。以下的方法有助于发现程序中的这类逻辑错误：

对照你的算法流程图，试着将你的大脑比作一台计算机，将你所编写的程序代码一行一行地在头脑中执行，不要漏掉任何一行，也不要试图跳过任何一部分代码。在遇到程序流程发生分支的地方，要对每一条分支都进行“执行”检查，并记录下程序的流程线路。

经过上述方法的检查，纠正了程序中的逻辑错误后，将程序输入计算机中，借助代码编译器的提示，修改程序中未发现的语法错误并调试程序直至通过。

1.2 Windows 的工作方式

Windows 的工作方式：窗口、事件和消息。

全面地讨论 Windows 的内部工作机制将需要整整一本书的容量，我们没有必要深入了解所有的技术细节。Windows 的工作机制，简单地说就是三个关键的概念：窗口、事件和消息。

不妨简单地将窗口看作带有边界的矩形区域。也许已经了解几种不同类型的窗口：如 Windows 98 的“资源管理器”窗口、文字处理程序中的文档窗口或者弹出提示有关信息的对话框。除了这些最普通的窗口外，实际上还有许多其它类型的窗口。命令按钮是一个窗口，图标、文本框、选项按钮和菜单条也都是窗口。

Microsoft Windows 操作系统通过给每一个窗口指定一个唯一的标识号（窗口句柄 `hWnd`）来管理所有的窗口。操作系统连续地监视每一个窗口的活动或事件的信号。事件可以通过诸如单击鼠标或按下按键的操作而产生，也可以通过程序的控制而产生，甚至可以由另一个窗口的操作而产生。

每发生一次事件，都将引发一条消息发送至操作系统。操作系统处理该消息并广播给其它窗口。然后，每一个窗口根据自身处理该条消息的指令而采取适当的操作（例如，当窗口解除了其它窗口的覆盖时，重显自身窗口）。

可以想象，处理各种窗口、事件和消息的所有可能的组合将有惊人的工作量。幸运的是，Visual Basic 使用户摆脱了所有的低层消息处理。许多消息由 Visual Basic 自动处理了，其它的作为事件过程由编程者自行处理。这样可以快速创建强大的应用程序而无需涉及一些不必要的细节。

1.3 Windows 面向对象的事件驱动程序设计

使用过 Windows 的用户都知道，用户在程序运行的过程中可以通过键盘或使用鼠标单击按钮、操作滚动条和从列表框或菜单项选择来提供输入，以完成与程序的一个交互式对话，用来控制程序的流程。因而，就要求用于编写 Windows 应用程序的编程语言必须能够创立 Windows 环境中的各种标准对象，如各类按钮、滚动条、列表框、复选框、菜单条等等。Visual Basic 是专门为开发 Windows 应用程序而设计的一种编程语言，它支持用户实现面向对象的事件驱动程序编程。

谈到面向对象的编程技术（OOP），也许大家比较陌生，但这是开发 Windows 应用程序非常关键的一种技术，它的核心是可以创建一些可重用的对象，并使用这些对象来建立程序。举个例子来说，在各类 Windows 应用程序中大量使用的命令按钮（“确定”按钮、“取消”按钮等）就是一种可重用的 Windows 标准对象。在开始编程时，先创建好这样的—个标准对象（当然，Visual Basic 已经为用户完成了这部分工作），并且定义好这个对象的属性（如按钮的形状、颜色，按钮中字体的类型、大小等）、方法（指对象对特定事件作出响应的方法，如按钮被鼠标单击后应完成的动作）和可响应的事件（指由用户或操作系统所引发的动作，对于按钮来说有单击鼠标或双击鼠标等事件）等特性。然后在创建程序时，根据程序的需要，将这种标准的按钮插入到当前程序界面中，并适当修改按钮的上述几个特性，即可完成一段程序模块的创建。Windows 应用程序就是由许多这样的程序模块所组成的。

说到这里，也就引出了 Windows 编程的另一项重要技术——事件驱动程序编程。我们先来回顾一下 DOS 平台下程序的运行过程。

在出现 Windows 之前，程序以顺序方式运行。也就是说，一旦程序启动，它就一条指令—条指令地向前执行，直至程序结束或出现致命错误。运行程序通常的步骤是：

- 创建输入文件；
- 启动程序；
- 等待程序运行直至程序结束；
- 检查输出文件或打印报告。

这就是传统的“过程化”的应用程序。应用程序自身控制了执行哪一部分代码和按何种顺序执行代码。虽然它们也能完成用户所要求的任务，但不如今天的程序易于使用。但是这些程序易于编写，这是因为每个程序都有明确定义的任务，而且几乎没有与用户的交