

● 赵胜利 编著

Visual Basic 6.0

for Windows

编程实用手册

人民邮电出版社



计算机技术丛书

Visual Basic 3.0 for Windows

编程实用手册

赵胜利 编著

人民邮电出版社

丛 书 前 言

世界上发达国家普遍重视发展以计算机和通信为核心的信息技术、信息产业和信息技术的应用,一些经济发达国家信息产业发展迅速。

当前,我国处于国民经济高速发展时期。与此相伴随,必将有信息技术、信息产业和信息技术应用的高速发展。各行各业将面临信息技术应用研究与发展的大课题以及信息化技术改造的大任务、大工程。

为了适应信息技术应用大众化的趋势,提高应用水平,我们组织编写、出版了这套“计算机技术丛书”。这套丛书以实用化、系列化、大众化为特点,介绍实用计算机技术。

这套丛书采取开放式选题框架,即选题面向我国不断发展着的计算机技术应用的实际需要和国际上的实用新技术,选题不断增添又保持前后有序。

这套丛书中的著作还拟配合出版软件版本,用软盘形式向读者提供著作中介绍的软件,以使读者方便地使用软件。

我们希望广大读者为这套丛书的出版多提意见和建议。

前 言

编写 Windows 环境下的应用软件,无论是对专业程序员还是计算机爱好者来说,都是一件很有吸引力的事情。

用 C++ 之类的语言工具去开发 Windows 应用程序,不是件容易的事。以往,在 DOS 下写两三行代码就能解决问题的程序,移植到 Windows 环境下,至少要写两三页代码。编写 Windows 环境下的程序代码,还面临许多要重新学习的东西:控制字体、建立窗口和菜单、安排内存、使用系统资源、熟悉六百多个接口函数、弄懂同时运行的程序间的数据交换方法、转向事件驱动的方式的编程思路等。一般情况下,用 C++ 编写 Windows 下的应用程序,先要学会使用 Microsoft 公司提供的 Windows 应用程序开发工具 SDK,这就要花费很多的时间和精力。

人们希望有一种语言工具,在 Windows 下编程就像在 DOS 下使用 Quik Basic 编程那样简便。应运而生的这种工具就是微软公司 1991 年推出的 Visual Basic,字面意思是“可视的 BASIC”。我们这里指的是国内大量使用的 Visual Basic 3.0。Visual Basic 体现了许多先进的编程思想,它的方法新鲜奇特,用它很容易编写出专业化的 Windows 应用软件,而且编写的软件在运行速度上丝毫不逊色于用 C++ 编写的软件。它的编程效率和编程的简易性受到一致的赞美。用过 Visual Basic 的人都感到,用它编写 Windows 应用程序比用 Quik Basic 编写 DOS 应用程序还要容易。这种工具相当完美,不论是书写程序还是调试软件,都非常简单。这使它成为公认的编写 Windows 应用软件的最好用、最优秀的编程环境之一。

Visual Basic 的另一个为人称道的优点是特别好学,使用方便。研究如何用 C++ 编写程序可能需要几个月的时间,但当你借助一本好的教材学 Visual Basic 时,发现只用一两天便入了门,一个月以后已经可以编写非常专业化的软件了。

作为一种新的语言工具,它的特点不是三言两语能说得清的事情。本书是一本完整、简明的编程技术参考书,提供尽可能完整的语言参考资料,以便读者尽快了解和建立新的编程思想,熟悉这一优美的编程环境,编写出漂亮的、专业化的应用软件。

本书共分 10 章和 6 个附录。前 8 章对 Visual Basic 编程环境和编程技术的各个方面作了提纲式的简明介绍,并辅助以简单的举例。在第 9 章介绍了一个图形编程的完整例子,在第 10 章介绍了一个通信方面的实用的例子,帮助读者消化前 8 章所学的内容,并可以模仿例子进行编程。前 5 个附录分别按字典形式列出了 Visual Basic 的属性、方法、事件、语句和函数,便于读者在学习和编程时查找使用。第 6 个附录列出了错误信息号码和它的中文解释,便于在调试程序时使用。

这本书里提供的例子,都经过上机验证(也在 Windows 95 中文测试版下试验过,但界面颜色和字形要修改)和实际运行的考验。

本书使用的是专业版 Visual Basic,但讲解的内容以标准版为主。

在此要特别感谢北京航空航天大学的庞宏冰先生以及他的同事,庞先生慷慨地允许发表由他主持开发的一个正在电力系统生产现场使用着的有关通信的应用程序,来作为这本书的第10章。

由于作者水平有限,时间仓促,书中错误之处在所难免,请读者批评指正。

作 者

目 录

第一章 预备知识	1
1.1 Windows 应用程序	1
1.2 Windows 环境下的程序运行机制	2
1.3 面向对象编程	3
1.3.1 基本概念	3
1.3.2 对象的属性	4
1.3.3 对象的方法	4
1.4 Visual Basic 中的面向对象的编程方法	5
1.4.1 可视的程序设计环境	5
1.4.2 VB 所能识别的对象	5
1.4.3 VB 对象的属性、方法和事件	6
1.5 VB 的语句、函数和过程	6
1.5.1 程序的语句和函数	6
1.5.2 基本语句	6
1.5.3 程序过程	7
1.5.4 自定义函数	7
1.5.5 引用算子“.”和“_”	7
1.5.6 约定的印刷格式	7
1.6 VB 的基本数据类型和运算符	8
1.6.1 基本数据类型	8
1.6.2 VB 的运算符	10
1.7 VB 的项目	11
1.8 VB 对硬件及软件环境的要求	12
1.9 安装并进入 VB 环境	12
第二章 VB 的编程环境	14
2.1 主(Main)窗口	14
2.1.1 功能和组成	14
2.1.2 标题栏	14
2.1.3 主菜单	15
2.1.4 工具栏	28
2.2 窗体窗口(Form)	29
2.3 项目窗口(Project)	30
2.3.1 功能和组成	30
2.3.2 标题	30

2.3.3 View Form 按钮	30
2.3.4 View Code 按钮	30
2.4 属性窗口(Properties)	31
2.4.1 功能及组成	31
2.4.2 标题	31
2.4.3 对象名称下拉表	31
2.4.4 属性编辑框	31
2.4.5 属性列表框	31
2.5 工具箱窗口(ToolBox)	31
2.5.1 功能和组成	31
2.5.2 标题条	32
2.5.3 控件工具图标	32
2.5.4 控件的名称	32
2.6 其他窗口	32
2.6.1 编码窗口(Code)	32
2.6.2 调色板窗口(Color Palette)	33
2.6.3 调试窗口(Debug)	35
2.6.4 菜单设计窗口(Menu Design)	35
第三章 VB 编程基础	38
3.1 如何尽快入门	38
3.2 建立一个简单的应用程序	38
3.2.1 建立 VB 程序基本步骤	38
3.2.2 设计用户界面和设置属性	39
3.2.3 编写事件过程	40
3.2.4 试运行程序	42
3.2.5 生成 EXE 文件	43
3.2.6 打印程序清单	43
3.3 VB 的常量和变量	45
3.3.1 常量	45
3.3.2 变量	46
3.4 VB 的数组	49
3.4.1 数组	49
3.4.2 动态数组	50
3.4.3 控件数组	50
3.4.4 数组和对象	51
3.4.5 数组的清除	51
3.5 VB 的程序控制结构	51
3.5.1 三种控制结构	51
3.5.2 分支语句	51
3.5.3 循环语句	53

3.6 VB的过程和函数	54
3.6.1 事件过程	54
3.6.2 通用过程	54
3.6.3 自定义函数	56
3.6.4 参数的址传递和值传递	57
3.7 作用域	60
3.7.1 VB程序块的作用范围	60
3.7.2 常量、变量和数组的作用域	60
3.8 运行时错误的捕获和处理	62
3.8.1 运行时错误	62
3.8.2 捕获和处理	62
第四章 窗体和控件(一)	64
4.1 窗体(Form)对象	64
4.1.1 什么是窗体	64
4.1.2 自定义坐标系	64
4.1.3 窗体的属性	64
4.1.4 窗体的方法	65
4.1.5 窗体的事件	66
4.1.6 例子1:使用多个窗口,在过程中设置属性值	66
4.1.7 例子2:AutoRedraw 属性,Paint 事件,Main 过程	75
4.2 标签(Label)控件	78
4.2.1 什么是标签	78
4.2.2 标签的属性	78
4.2.3 标签的方法	79
4.2.4 标签的事件	79
4.2.5 例子1:使用属性设置窗口,Click 事件,控件数组	79
4.2.6 例子2:鼠标拖动控件,方法 Drag	89
4.3 文本框(Text Box)控件	91
4.3.1 什么是文本框	91
4.3.2 文本框的属性	91
4.3.3 文本框的方法	92
4.3.4 文本框的事件	92
4.3.5 例子:Change 事件	92
4.4 命令按钮(CommandButton)控件	96
4.4.1 什么是命令按钮	96
4.4.2 命令按钮的属性	96
4.4.3 命令按钮的方法	96
4.4.4 命令按钮的事件	96
4.4.5 例子:热键的设置	97
4.5 单选按钮(OptionButton)控件	99

4.5.1 什么是单选按钮	99
4.5.2 单选按钮的属性	99
4.5.3 单选按钮的方法	99
4.5.4 单选按钮的事件	100
4.5.5 例子:属性值 Value	100
4.6 复选框(CheckBox)控件	104
4.6.1 什么是复选框	104
4.6.2 复选框的属性	104
4.6.3 复选框的方法	105
4.6.4 复选框的事件	105
4.6.5 例子	105
4.7 分类框(Frame)控件	112
4.7.1 什么是分类框	112
4.7.2 分类框的属性	112
4.7.3 分类框的方法	112
4.7.4 分类框的事件	113
4.7.5 例子	113
4.8 滚动条(ScrollBar)控件	119
4.8.1 什么是滚动条	119
4.8.2 滚动条的属性	120
4.8.3 滚动条的方法	120
4.8.4 滚动条的事件	120
4.8.5 例子:属性值 Value,Max,Min	120
4.9 直线(Line)控件	124
4.9.1 什么是直线	124
4.9.2 直线的属性	124
4.9.3 直线的方法	124
4.10 外形(Shape)控件	125
4.10.1 什么是外形	125
4.10.2 外形的属性	125
4.10.3 外形的方法	125
4.11 计时器(Timer)控件	125
4.11.1 什么是计时器	125
4.11.2 计时器的属性	125
4.11.3 计时器的事件	126
4.11.4 例子:Timer 事件和 InterVal 属性	126
4.12 图片框(PictureBox)控件	129
4.12.1 什么是图片框	129
4.12.2 图片框的属性	130
4.12.3 图片框的方法	131

4.12.4 图片框的事件	131
4.12.5 例子:属性 Align 和 AutoSize	131
4.13 图像(Image)控件	133
4.13.1 什么是图像	133
4.13.2 图像的属性	133
4.13.3 图像的方法	134
4.13.4 图像的事件	134
4.13.5 例子:属性 Stretch	134
4.14 菜单的设计	136
4.14.1 什么是菜单	136
4.14.2 菜单的属性	136
4.14.3 菜单的事件	136
4.14.4 菜单的例子	137
第五章 窗体和控件(二)	140
5.1 MDI 窗体	140
5.1.1 MDI 窗体的概念	140
5.1.2 MDI 窗体的属性	140
5.1.3 MDI 窗体的方法	141
5.1.4 MDI 窗体的事件	141
5.1.5 多文档界面的例子	141
5.2 列表框(ListBox)控件	145
5.2.1 什么是列表框	145
5.2.2 列表框的属性	145
5.2.3 列表框的方法	145
5.2.4 列表框的事件	146
5.2.5 列表框的使用	146
5.2.6 例子:将表项从一个列表框移向另一个列表框	146
5.3 组合框(ComboBox)控件	149
5.3.1 什么是组合框	149
5.3.2 组合框的属性	149
5.3.3 组合框的方法	150
5.3.4 组合框的事件	150
5.3.5 组合框的用法	150
5.3.6 组合框的例子	150
5.4 驱动器列表框(DriveListBox)控件	153
5.4.1 什么是驱动器列表框	153
5.4.2 驱动器列表框的属性	153
5.4.3 驱动器列表框的方法	154
5.4.4 驱动器列表框的事件	154
5.5 目录列表框(DirectoryListBox)控件	154

5.5.1 什么是目录列表框	154
5.5.2 目录列表框的属性	154
5.5.3 目录列表框的方法	155
5.5.4 目录列表框的事件	155
5.6 文件列表框(FileListBox)控件	155
5.6.1 什么是文件列表框	155
5.6.2 文件列表框的属性	155
5.6.3 文件列表框的方法	156
5.6.4 文件列表框的事件	156
5.6.5 文件列表框的例子	156
5.7 通用对话框(CommonDialog)控件	162
5.7.1 什么是通用对话框	162
5.7.2 通用对话框的一般属性	162
5.7.3 文件打开(Open)对话框及其特殊属性	162
5.7.4 文件打开对话框的例子	162
5.7.5 文件存盘(Save As)对话框及其特殊属性	164
5.7.6 文件存盘对话框的例子	164
5.7.7 字体选择(Font)对话框及其特殊属性	165
5.7.8 字体选择对话框的例子	166
5.7.9 打印(Print)对话框及其特殊属性	166
5.7.10 打印对话框的例子	166
5.7.11 调色板(Color)对话框及其特殊属性	167
5.7.12 调色板(Color)对话框的例子	168
5.7.13 帮助(Help)对话框及其特殊属性	169
5.7.14 帮助对话框的例子	169
5.8 网格(Grid)控件	171
5.8.1 什么是网格	171
5.8.2 网格的属性	171
5.8.3 网格的方法	172
5.8.4 网格的事件	172
5.8.5 网格的使用	172
5.9 数据(Data)控件	172
5.9.1 什么是数据控件	172
5.9.2 数据控件的属性	173
5.9.3 数据控件的方法	173
5.9.4 数据控件的事件	174
5.9.5 数据控件的例子	174
5.10 数据约束控件	176
第六章 文件管理和数据库管理	183
6.1 文件的输入和输出	183

6.1.1 文件管理系统	183
6.1.2 顺序文件的访问	184
6.1.3 随机文件的访问	185
6.1.4 二进制文件的访问	186
6.1.5 与文件操作有关的语句和函数	187
6.1.6 例子:一个简单的文件处理器	187
6.2 对象变量	191
6.2.1 对象变量的创建	191
6.2.2 对象变量的 Count 属性	193
6.2.3 关键字 Me 与对象实例	193
6.2.4 例子:符号!	193
6.3 数据库管理对象	195
6.3.1 VB 对数据库的理解	195
6.3.2 字段(Field)对象	196
6.3.3 字段集(Fields)对象	197
6.3.4 索引(Index)对象	197
6.3.5 索引集(Indexes)对象	198
6.3.6 表定义(TableDef)对象	198
6.3.7 表定义集(TableDefs)对象	198
6.3.8 查询定义(QueryDef)对象	199
6.3.9 表(Table)对象	199
6.3.10 动态记录集(Dynaset)对象	200
6.3.11 静态记录集(Snapshot)对象	200
6.3.12 数据库(Database)对象	201
6.4 创建和打开数据库	202
6.4.1 创建一个数据库	202
6.4.2 例子:函数 CreateDatabase()	202
6.4.3 打开一个数据库	203
6.4.4 生成一个表(Table)	204
6.4.5 生成一个动态记录集(Dynaset)	204
6.4.6 生成一个静态记录集(Snapshot)	204
6.4.7 生成一个附加表(Attached Table)	205
6.5 数据库的操作	205
6.5.1 定位记录	205
6.5.2 增加记录	205
6.5.3 删除记录	206
6.5.4 修改记录	206
6.5.5 属性的缺省写法	206
6.5.6 根据条件生成新的记录集	206
6.6 复习:FoxPro 2.5 数据浏览器	206

第七章 应用程序间的数据交换	323
7.1 程序间数据交换的类型	323
7.2 使用剪贴板(Clipboard)交换数据	323
7.2.1 剪贴板和它的方法	323
7.2.2 使用剪贴板的例子	324
7.3 使用 DDE 交换数据	326
7.3.1 什么是 DDE	326
7.3.2 建立链接最直接的方法	327
7.3.3 通过控件属性在程序中建立链接	327
7.3.4 DDE 的方法	328
7.3.5 DDE 过程中的事件	329
7.3.6 DDE 的例子	329
7.4 对象链接与嵌入(OLE)	337
7.4.1 什么是 OLE	337
7.4.2 OLE 和 DDE 的区别	338
7.4.3 OLE 控件的属性	339
7.4.4 OLE 的方法	339
7.4.5 OLE 的事件	339
7.4.6 在设计阶段包容一个对象	340
7.4.7 在程序运行阶段包容对象	340
7.4.8 例子:OLE 控件的使用	342
第八章 充分利用系统资源	355
8.1 VB 可用的系统资源	355
8.2 动态链接库(DLL)	356
8.2.1 什么是 DLL	356
8.2.2 在 VB 程序中调用 DLL 函数	356
8.3 制作应用程序安装文件	357
8.4 联机帮助(Help)文件的制作	357
8.4.1 什么是 Help 文件	357
8.4.2 Help 文件的结构	358
8.4.3 编写 Help 源文件的方法	358
8.4.4 编写 Help 项目文件的方法	360
8.4.5 编译生成 Help 文件	364
第九章 一个完整的图形程序	365
9.1 图形程序:图标制作	365
9.2 设计阶段的窗体安排	370
9.3 全局声明和常量定义	372
9.4 通用过程	381
9.5 编辑窗口的窗体安排和编程	392
9.6 观察窗体的安排和编程	442

9.7 调色板窗口的窗体安排和编程	460
9.8 “关于”窗口的窗体安排和编程	470
第十章 通信程序实例	474
10.1 关于例子的说明	474
10.2 通信(MSCOMM)控件简介	474
10.3 关于调制解调器的基本命令	475
10.4 XModem 通信协议	477
10.5 程序清单	478
附录 1 VB 属性的说明及解释	
附录 2 VB 方法的说明及解释	
附录 3 VB 事件的说明及解释	
附录 4 VB 语句的说明及解释	
附录 5 VB 函数的说明及解释	
附录 6 VB 错误信息的说明	

第一章 预备知识

1.1 Windows 应用程序

在微型计算机操作系统 MS-DOS 下运行的 Microsoft Windows 是一个基于图形用户界面的多任务、多窗口操作环境。Microsoft Windows 特别友好的用户界面和与设备无关的图形设备接口,已经被程序员和用户所接受,成为事实上的图形操作环境标准。

这里所说的 Windows 应用程序,是在 Windows 环境下运行的扩展名为 .EXE 的可执行文件。它有以下主要特点:

1. 有专门的格式,只能在 Windows 环境下运行。
2. 有一个矩形窗口。窗口的大小和位置,可以根据程序或用户的需要随时改变,既可以扩展至整个屏幕,也可以缩成一个图标。
3. 程序和用户的对话,遵循 Windows 的标准用户界面使用方法。
4. 在 Windows 环境下,可以同时运行多个 Windows 应用程序,即执行多个任务,用户可以选择(激活)一个任务窗口进行操作。每个任务可以按前台、后台或独占的方式运行。
5. 在多任务运行时,必要时几个任务可以执行同一个程序,而且各个任务的输入输出内容可以不同步。
6. 同时执行的两个任务间可以通过使用剪贴板的静态数据交换,或者 DDE(动态数据交换)和 OLE(对象链接与嵌入)进行通信和共享数据。
7. Windows 环境为所有它支持的外部设备提供驱动程序,即提供与设备无关的图形设备接口。
8. 可使用计算机装入的所有内存,并把硬盘的一部分作为虚拟内存使用。
9. 使用基于图形学的 True Type 字形显示技术,使屏幕显示和打印效果达到最大的一致性:“所见即所得”(WYSIYG)。
10. 各个运行程序实质上受 Windows 管理系统的调度和控制,这是 Windows 的核心软件,它支配和管理着 Windows 环境的运行。

以上特点带给用户许多方便:

1. 操作界面是标准的,并且模拟现实生活,所以特别简单直观,只需用键盘和鼠标对图标、对话框、菜单、按钮等标准界面元素进行选择 and 操作就可完成程序对话。用户只要学会使用一种软件,就能很轻松地学会使用所有的软件。
2. 不必为每个应用程序建立专门的外设驱动程序。
3. 多任务可以使用户不必再考虑将一些附加程序常驻内存,而且随时可以进行程序间的数据互访和通信。

4. 不必再担心内存不够的问题。
5. 使用上比较自由和随意。

对编程人员来说,好处是:

1. 一般不再需要为外部设备的配置及外设驱动程序的编写伤脑筋。
2. 使用标准的图形操作界面,减轻了为设计界面及用户操作方法所带来的沉重负担。
3. Windows 的 API(应用过程接口)有大量丰富的标准化了的图形例程可供编程使用。
4. 编程不像 DOS 下那么拘泥(下面要解释)。

但是,大量新概念的引入及 600 多个 API 函数,使过去在 DOS 下工作的编程人员对如何编好 Windows 应用程序感到困惑和信心不足。因此,反而使编程人员感到不方便。

1.2 Windows 环境下的程序运行机制

从编写 DOS 应用程序转到编写 Windows 应用程序,需要一些观念上的转变。其中最重要的一个转变是:从编写 DOS 下**顺序的、过程驱动**的程序,转变到编写 Windows 下**随机的、事件驱动(Event-Driven)**的程序。

传统的 DOS 下的应用程序与用户的“感情距离”较远,原因是 DOS 应用程序是顺序的、过程驱动的:有一个明确的开始,明确的过程和一个明确的结束。按程序要求的顺序,先干什么,后干什么,次序不能颠倒,否则程序将提示出错或陷入混乱。因此,用户在使用程序时是被动的,是程序的奴隶。而程序员在编程时,要精确安排好各个程序过程的先后顺序:开始运行哪个模块,先提示用户干什么,接着程序该干什么,等等。所有的细节都不能疏忽。如果我们平时的工作也这么机械,不能越雷池一步,同样会感到很不自在。

事实上,通常的工作方法是,在写一份报告时突然接到电话,在通电话的过程中,又要根据电话询问去查一件资料,干累了还要找同事聊聊天,甚至拒绝干自己不喜欢干的事情,或者突然感到不舒服要休息,等等。也就是说,“突发”的事件和你的“求自由”的本能,使你不可能按部就班地工作。你要马上对这些突发事件做出反应,进行相应的处理。因而,我们自然希望和计算机的对话也应当这样自然。Windows 环境下程序的运行恰好照顾到了这一点,这就是它的“事件驱动”的程序运行机制。

Windows 下的应用程序是事件驱动的:由一些不可能知道先后顺序的随机发生的事件来控制与程序的对话。这些事件的识别和处理由 Windows 的管理系统来安排。

Windows 应用程序的事件驱动方式,使应用程序和用户的“感情”拉近了。因为它的程序运行方式,是建立在应付种种突发“事件”的机制上的,用户使用起来比较自由和随意。

Windows 是多任务运行环境,可以同时运行几个程序。我们可以在运行一个程序时,突然跳到另一个程序上去查询某些数据。所以,程序采用事件驱动的方式是很自然的。这更接近于我们人的思维活动和工作方式。

Windows 中的事件,是指能被 Windows 管理系统感知的运行状态的变化:外部设备输入的信息(例如:按了一下鼠标按钮),系统内部的某种信息转移(例如:时钟运行的时间长度达到了预设值)。

由外部设备(如键盘)产生的事件叫**外部事件**,由系统内部(如系统时钟)产生的事件叫**系**

统事件。

在程序运行中,用户可以心血来潮,“制造”一起事件,而程序必须判明这是一起什么事情,应该作出什么样的响应。在程序运行过程中,用户是主动的,是主人。

在 Windows 环境下,编程人员的任务是,决定程序能感知什么样的事件,并对事件做出什么样的反应。至于程序运行过程中事件何时发生,发生哪种事件,由 Windows 管理系统管理,与程序员无关。

注意:事件驱动程序的运行机制和 DOS 下的中断方式的程序运行机制不同。根本区别在于,事件驱动程序中的事件是由 Windows 管理系统按“时间分割”的方式统一管理的,没有哪一个应用程序可以任意单独占用中央处理器时间。这是由 Windows 的多任务运行方式决定的。而在 DOS 下当一个应用程序在处理中断时,它将独占处理器时间。

事件驱动程序的运行机制,过程大致如下:

1. 等待,监视:Windows 管理系统监视着事件的发生:每发生一个事件,就将其放入管理系统的一个专门的队列中。

2. 接收,排队:Windows 管理系统判断进入专门队列的事件,是由用户或系统对哪个应用程序发出的,从而将该事件拷贝到相应的应用程序的事件队列中。

3. 分析,响应:应用程序分析进入该应用程序事件队列的事件信息,根据事件信息内容调用一段专门处理该事件的程序代码。在执行完程序代码后,应用程序继续分析下一个进入队列的事件信息,调用相应的程序代码。这种程序代码通常由编程人员编写,叫**事件过程**。

Windows 管理系统本身的任务就是不断对事件进行等待、监视、接收、排队、分析和响应。在事件驱动程序的编写过程中,编程人员的主要任务是对每一可能发生的事件编写出处理这一事件的程序代码,即编写事件过程。

1.3 面向对象编程

1.3.1 基本概念

用传统的方法编写 Windows 应用程序,并没有给编程人员带来多大益处。沿用传统的办法编写 Windows 应用程序,想要提高效率,几乎是行不通的。

现在,编写这种复杂庞大的程序,至少要用**面向对象(OOP)**的编程方法。下面先简单介绍一下面向对象编程的概念。

首先,使用 Windows 的图形用户界面,类似于操作一台真实的机器。可以通过鼠标这只“手”,按动屏幕上所画(模拟机器)的按钮,或通过键盘这只“笔”,在屏幕上的对话框内写一段文字,指示机器该干什么。这些按钮和对话框,在 Windows 的各个应用程序中,用法基本一样,外形(除具体尺寸、颜色)也大同小异。这就和工厂组装机器的标准件一样,编程人员也可以用这些“标准件”组装出一个用户界面来。凡是用户可以访问的这些“标准件”,用术语说就是**对象(Object)**。在面向对象的编程中,这些对象成为编程语言的一些基本元素,不再需要对它们编写代码。

进一步说,用面向对象的编程方法编写 Windows 应用程序的主要思想是,有关窗口、对话