

Visual BASIC 5.0 系列丛书

魏源源 主编
李纪鸿 唐磊 等编

Visual BASIC 5.0

中文版程序设计教程

Visual BASIC 5.0



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.com.cn>

7/312
WY/1

Visual Basic 5.0

中文版程序设计教程

魏源源 主编
李纪鸿 唐 磊 等编

電子工業出版社
Publishing House of Electronics Industry

内 容 提 要

Visual Basic 由于功能强大、简单易用而有着广泛的用户,尤其是在数据库、多媒体、商用系统的用户界面以及一般应用程序的设计等方面,Visual Basic 更是有其他开发环境所不能取代的优越之处。升级到 5.0 版后,其功能和性能都得到了增强,而汉化后的 Visual Basic 5.0 中文版,由于有全新的中文界面,更是方便了用户的使用。

本书系统、全面地讲述了 Visual Basic 5.0 中文版的使用方法。全书共分十一章和一个附录,内容如下:中文 Visual Basic 5.0 基础知识,窗体、控件与菜单设计,工程的使用,Visual Basic 5.0 语言简介,界面设计,标准控件的使用,对象的使用,部件的使用,鼠标与键盘事件的处理,文本与图形处理,数据库操作等。附录中还介绍了 Visual Basic 5.0 的安装方法与 IDE 的使用。

本书内容全面,循序渐进,范例丰富,是广大 Visual Basic 用户不可多得的参考书。

JS255/17

书 名:Visual Basic 5.0 中文版程序设计教程

主 编:魏源源

编 者:李纪鸿 唐 磊 等

责任编辑:胡毓坚

特约编辑:宋 岩

印 刷 者:北京天竺颖华印刷厂

装 订 者:三河市金马印装有限公司

出版发行:电子工业出版社出版、发行

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话 68214070

URL:<http://www.phei.com.cn>

经 销:各地新华书店经销

开 本:787×1092 毫米 1/16 印张:28.25 字数:684 千字

版 次:1998 年 3 月第 1 版 1998 年 6 月第 2 次印刷

书 号:ISBN 7-5053-4616-4
TP·2194

定 价:34.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换
版权所有·翻印必究

前 言

Visual Basic 由于功能强大、简单易用而有着广泛的用户,尤其是在数据库、多媒体、商用系统的用户界面以及一般应用程序的设计等方面,Visual Basic 更是有其他开发环境所不能取代的优越之处。升级到 5.0 版后,其功能和性能都得到了增强,而汉化后的 Visual Basic 5.0 中文版,由于有全新的中文界面,更是方便了用户的使用。

Visual Basic 5.0 有三种版本,各自满足不同的开发需要:“Visual Basic 学习版”使编程人员轻松开发 Windows 95 和 Windows NT 应用程序;“专业版”为专业编程人员提供了一整套进行开发的功能完备的工具;“企业版”使得专业编程人员能够开发功能强大的分布式应用程序。本书以三种 Visual Basic 5.0 版本的公共方面为主题,系统、全面地讲述了 Visual Basic 5.0 中文版的使用方法。全书内容如下:中文 Visual Basic 5.0 基础知识,窗体、控件与菜单设计,工程的使用,Visual Basic 5.0 语言简介,界面设计,标准控件的使用,对象的使用,部件的使用,鼠标与键盘事件的处理,文本与图形处理,数据库操作,如何创建安装程序等。附录中还介绍了 Visual Basic 5.0 的安装方法与 IDE 的使用。

本书是一本容易理解、对 Visual Basic 5.0 的几乎所有方面(除错误处理之外)都有详尽说明的手册,并且本书假定读者以前从未用 Visual Basic 编过程序。对于那些以前使用过 Visual Basic 的早期版本或其他结构化语言编过程序的读者来说,本书当然会更容易掌握。如果从头开始学习,那么很快就能熟练掌握 Visual Basic 5.0,并能充分利用其强有力的事件驱动特性去编写精彩、完美的 Windows/Windows NT 应用程序。读完本书之后,读者就可以用 Visual Basic 编写商品化的程序了。

本书内容庞大,由多人合作而成,除署名作者外,参与编写的还有王仲谋、施凤霞、朱志勇、李宁、李华、农军、杨跃、张贵军、朱莉、朱志坚、李梅。刘海燕和魏宁对全书进行了审校。张静、张群负责全书的资料收集整理工作,陈群、狄文对本书的录入工作付出了艰苦地劳动。

目 录

第一章 中文 Visual Basic 5.0 基础	(1)
1.1 Visual Basic 概念	(1)
1.2 第一个 Visual Basic 应用程序	(2)
1.2.1 创建界面	(2)
1.2.2 设置属性	(4)
1.2.3 编写代码	(5)
1.2.4 运行应用程序	(7)
第二章 窗体、控件与菜单	(9)
2.1 设计窗体	(9)
2.1.1 设置窗体属性	(9)
2.1.2 窗体事件和方法	(10)
2.2 单击按钮执行操作	(10)
2.2.1 使用 CommandButton	(11)
2.2.2 “测试”按钮应用程序	(11)
2.3 显示和输入文本的控件	(12)
2.3.1 用 Label 显示文本	(12)
2.3.2 使用 TextBox	(13)
2.4 为用户提供选择的控件	(15)
2.4.1 用 CheckBox 选择单个选项	(15)
2.4.2 用选项按钮进行选项分组	(17)
2.4.3 使用 ListBox 和 ComboBox	(20)
2.4.4 使用 ScrollBar 作为输入设备	(21)
2.5 显示图片和图形的控件	(21)
2.5.1 使用 PictureBox 控件	(22)
2.5.2 轻图形控件	(23)
2.5.3 Images 应用程序	(24)
2.6 其他控件	(25)
2.7 焦点	(27)
2.8 设置 Tab 键顺序	(27)
2.9 菜单基础	(28)
2.9.1 弹出式菜单	(29)
2.9.2 使用菜单编辑器	(29)
2.10 用对话框提示用户	(30)

第三章 工程	(33)
3.1 使用工程	(33)
3.1.1 工程资源管理器	(33)
3.1.2 工程文件	(34)
3.1.3 Visual Basic 工程的结构	(34)
3.2 创建、打开和保存工程	(35)
3.3 添加、删除和保存文件	(36)
3.4 在工程中添加控件	(37)
3.4.1 在工程中添加 ActiveX 控件	(37)
3.4.2 从工程中删除控件	(38)
3.4.3 使用其他应用程序的对象	(38)
3.4.4 使用资源文件	(39)
3.5 制作和运行可执行文件	(39)
3.6 设置工程选项	(40)
3.7 使用向导和外接程序	(41)
3.7.1 使用外接程序管理器	(41)
3.7.2 使用向导	(42)
第四章 关于 Visual Basic 5.0 语言	(45)
4.1 Visual Basic 应用程序的结构	(45)
4.1.1 事件驱动应用程序的工作方式	(46)
4.1.2 设计阶段	(46)
4.2 代码编写机制	(48)
4.2.1 代码模块	(48)
4.2.2 使用“代码编辑器”	(49)
4.2.3 编码基础	(50)
4.3 变量、常数和数据类型	(52)
4.3.1 变量	(52)
4.3.2 变量的范围	(55)
4.3.3 静态变量	(56)
4.3.4 常数	(57)
4.3.5 数据类型	(58)
4.3.6 数组	(64)
4.3.7 动态数组	(66)
4.4 过程	(68)
4.4.1 子过程(Sub 过程)	(68)
4.4.2 函数过程(Function 过程)	(70)
4.4.3 使用过程	(70)
4.4.4 向过程传递参数	(72)

4.5	控制结构.....	(76)
4.5.1	判定结构.....	(76)
4.5.2	循环结构.....	(78)
4.5.3	使用控制结构.....	(80)
4.6	对象.....	(82)
4.6.1	对象从何而来.....	(82)
4.6.2	用对象能做什么.....	(83)
4.6.3	使用对象初步.....	(83)
4.6.4	对象之间的关系.....	(86)
4.6.5	创建对象.....	(89)
4.7	利用多个工程进行工作.....	(95)
4.7.1	添加或删除工程.....	(95)
4.7.2	指定一个启动工程.....	(96)
4.8	管理应用程序设置值.....	(97)
4.8.1	创建或保存应用程序设置值.....	(97)
4.8.2	检索应用程序设置值.....	(98)
4.8.3	删除应用程序设置值.....	(99)
4.9	条件编译.....	(99)
4.10	资源文件.....	(101)
4.11	模板.....	(101)
4.12	命令行开关.....	(102)
4.13	将工程编译成本地代码.....	(103)
4.14	枚举.....	(105)
第五章	界面设计.....	(109)
5.1	多文档界面(MDI)应用程序	(110)
5.1.1	创建 MDI 应用程序.....	(111)
5.1.2	加载 MDI 窗体及其子窗体	(111)
5.1.3	维护子窗体的状态信息	(112)
5.1.4	用 QueryUnload 卸载 MDI 窗体.....	(112)
5.2	再论窗体	(113)
5.2.1	设置启动窗体	(113)
5.2.2	没有启动窗体时的启动	(113)
5.2.3	显示启动时的快速显示	(114)
5.2.4	结束应用程序	(115)
5.3	在应用程序中使用菜单	(115)
5.3.1	用菜单编辑器创建菜单	(116)
5.3.2	菜单标题与命名准则	(119)
5.3.3	创建子菜单	(120)

5.3.4	创建菜单控件数组	(120)
5.3.5	运行时创建和修改菜单	(121)
5.3.6	编写菜单控件的代码	(123)
5.3.7	显示弹出式菜单	(123)
5.3.8	MDI 应用程序中的菜单	(125)
5.4	工具栏	(127)
5.4.1	创建工具栏	(127)
5.4.2	协调菜单与工具栏的外观	(129)
5.5	对话框	(129)
5.5.1	模式与无模式的对话框	(129)
5.5.2	预定义对话框的使用	(130)
5.5.3	用窗体作为自定义对话框	(131)
5.6	各种显示类型的设计	(135)
5.6.1	设计与分辨率无关的窗体	(135)
5.6.2	设计不同浓度的颜色	(136)

第六章 标准控件..... (139)

6.1	Visual Basic 控件简介	(139)
6.1.1	控件分类	(139)
6.1.2	内部控件	(140)
6.1.3	标准 ActiveX 控件	(141)
6.1.4	更新旧版 Visual Basic 控件.....	(142)
6.1.5	控件命名约定	(142)
6.1.6	使用控件值	(143)
6.2	使用控件数组	(144)
6.2.1	为何使用控件数组	(144)
6.2.2	应用程序示例	(145)
6.2.3	设计时创建控件数组	(146)
6.2.4	运行时添加控件数组	(147)
6.2.5	在控件数组中添加和删除控件	(147)
6.3	使用 CheckBox 控件	(149)
6.3.1	Value 属性	(149)
6.3.2	Click 事件	(150)
6.3.3	响应鼠标和键盘	(151)
6.3.4	增强 CheckBox 控件的视觉效果	(151)
6.4	使用 ComboBox 控件	(151)
6.4.1	何时用组合框代替列表框	(151)
6.4.2	组合框的样式	(151)
6.4.3	添加项目	(152)

6.4.4	排序列表	(153)
6.4.5	删除项目	(154)
6.4.6	用 Text 属性获取列表内容	(154)
6.4.7	用 List 属性访问列表选项	(155)
6.4.8	用 ListIndex 属性判断位置	(155)
6.4.9	用 ListCount 属性返回项目数	(155)
6.5	使用 CommandButton 控件	(155)
6.5.1	向窗体添加命令按钮	(156)
6.5.2	选定命令按钮	(156)
6.5.3	增强命令按钮的视觉效果	(157)
6.6	使用 CommonDialog 控件	(157)
6.6.1	显示“打开”和“另存为”对话框	(158)
6.6.2	使用“颜色”对话框	(159)
6.6.3	使用“字体”对话框	(160)
6.6.4	使用“打印”对话框	(161)
6.6.5	使用 ShowHelp 方法显示帮助文件	(162)
6.7	使用 Data 控件	(163)
6.8	使用数据绑定控件(DBCombo、DBList 和 DBGrid)	(163)
6.8.1	DBList 和 DBCombo	(163)
6.8.2	DBGrid 控件	(163)
6.9	使用文件系统控件(DirListBox、DriveListBox 和 FileListBox)	(164)
6.9.1	检查文件系统	(164)
6.9.2	驱动器列表框	(165)
6.9.3	目录列表框	(165)
6.9.4	文件列表框	(167)
6.9.5	使用文件系统控件的组合	(168)
6.9.6	文件搜索器应用程序	(168)
6.10	使用 MSFlexGrid 控件	(171)
6.10.1	排序和合并数据	(172)
6.10.2	可进行单元编辑的工作表	(175)
6.11	使用 Frame 控件	(179)
6.12	使用 HScrollBar 和 VScrollBar 控件	(180)
6.12.1	Scroll Bar 控件如何工作	(180)
6.12.2	创建可滚动的图形视口	(181)
6.13	使用 Image 控件	(183)
6.14	使用 Label 控件	(185)
6.15	使用 Line 控件	(186)
6.16	使用 ListBox 控件	(187)
6.16.1	添加和删除项目	(191)

6.16.2 创建多列列表框.....	(194)
6.17 使用 OLE 容器控件	(196)
6.18 使用 OptionButton 控件	(197)
6.19 使用 PictureBox 控件.....	(198)
6.20 使用 Shape 控件.....	(200)
6.21 使用 TextBox 控件	(201)
6.22 使用 Timer 控件	(203)

第七章 对象..... (207)

7.1 Visual Basic 中的对象	(207)
7.1.1 找出对象所属的类	(207)
7.1.2 对一个对象执行多个动作	(207)
7.1.3 使用缺省属性	(208)
7.1.4 创建对象数组	(209)
7.1.5 创建对象集合	(211)
7.1.6 Visual Basic 集合对象	(211)
7.1.7 Visual Basic 中的集合	(215)
7.2 查找对象信息	(218)
7.2.1 对象浏览器的内容	(219)
7.2.2 添加对象的描述	(221)
7.2.3 在过程之间移动	(222)
7.2.4 浏览其他应用程序的对象	(222)
7.3 创建自己的类	(223)
7.3.1 类:将用户定义类型和过程组织在一起.....	(223)
7.3.2 对象:具有态的用户定义类型.....	(223)
7.3.3 定制窗体类	(224)
7.3.4 类模块的循序渐进	(227)
7.3.5 调试类模块	(230)
7.3.6 Visual Basic 窗体的存活期	(231)
7.3.7 类模块和标准模块的比较	(235)
7.4 向类中添加属性和方法	(237)
7.4.1 向类中添加属性	(238)
7.4.2 属性过程与公共变量的比较	(239)
7.4.3 运用属性过程	(240)
7.4.4 向类中添加方法	(244)
7.4.5 选择属性还是方法	(245)
7.4.6 使属性或方法成为缺省的	(246)
7.4.7 友元属性和方法	(247)
7.5 向类中添加事件	(248)

7.5.1	声明和引发事件	(249)
7.5.2	WithEvents 与窗体上的控件事件的比较	(254)
7.5.3	向窗体添加事件	(254)
7.5.4	声明、引发以及处理事件的总结	(256)
7.5.5	命名属性、方法和事件	(256)
7.6	多态	(256)
7.6.1	Visual Basic 是如何提供多态的	(257)
7.6.2	创建和实现接口	(258)
7.6.3	实现属性	(261)
7.6.4	关于对象和接口的简要补充讨论	(262)
7.6.5	代码重用的许多(内部)方面	(265)
7.7	对象引用和引用计数	(266)
7.8	对象模型	(267)
7.9	ActiveX 设计器	(271)
7.9.1	ActiveX 设计器	(271)
7.9.2	向“工程”菜单中添加 ActiveX 设计器	(274)
7.9.3	插入 ActiveX 设计器的新实例	(274)
第八章	部件	(277)
8.1	ActiveX 部件的类型	(277)
8.2	进程内的服务器和进程外的服务器	(278)
8.3	使用 ActiveX 部件	(278)
8.4	创建对对象的引用	(280)
8.4.1	二义性引用与引用优先权	(281)
8.4.2	浏览 ActiveX 部件类型库	(282)
8.4.3	声明对象变量	(283)
8.4.4	将对象引用赋予变量	(284)
8.5	使用对象的属性、方法和事件	(286)
8.5.1	使用对象的属性和方法	(286)
8.5.2	响应对象的事件	(287)
8.6	释放 ActiveX 部件	(288)
8.7	定位对象模型	(289)
8.7.1	定位对象层次	(289)
8.7.2	集合对象	(290)
8.7.3	使用外部可创建对象和从属对象	(291)
8.8	处理 ActiveX 部件中的运行时错误	(293)
8.8.1	测试对象引用	(294)
8.9	处理挂起 ActiveX 部件的请求	(294)
8.9.1	部件忙的状态	(294)

8.9.2	请求挂起状态	(295)
8.9.3	改变部件忙或请求挂起消息	(295)
8.9.4	控制超时间隔	(297)
8.9.5	产生部件忙超时错误	(298)
8.9.6	使用部件的可视界面	(299)
8.9.7	比较链接对象与嵌入对象	(300)
8.9.8	设计时用 OLE 容器控件插入对象	(301)
8.9.9	运行时让用户规定对象	(305)
8.9.10	决定在 OLE 容器控件中如何显示对象	(306)
8.9.11	激活 OLE 容器控件中对象	(307)
8.9.12	响应移动容器或调整容器大小	(307)
8.9.13	保存和检索嵌入的数据	(308)
第九章 鼠标与键盘事件		(311)
9.1	响应鼠标事件	(311)
9.1.1	MouseDown 事件	(311)
9.1.2	MouseMove 事件	(313)
9.1.3	MouseUp 事件	(314)
9.2	检测鼠标按钮	(315)
9.2.1	结合 MouseDown 和 MouseUp 事件使用 Button 参数	(316)
9.2.2	结合 MouseMove 使用 Button 参数	(317)
9.2.3	使用 Button 参数增强图形鼠标应用程序	(319)
9.3	检测 SHIFT、CTRL 及 ALT 键的状态	(320)
9.4	拖放	(321)
9.4.1	启动自动拖动模式	(322)
9.4.2	改变拖动图标	(322)
9.4.3	放下对象时的响应	(323)
9.4.4	启动拖动或停止拖动时的控制	(324)
9.4.5	改变控件的位置	(325)
9.5	OLE 拖放	(326)
9.5.1	启动自动 OLE 拖放	(328)
9.5.2	OLE 拖放 DataObject 对象	(329)
9.5.3	OLE 拖放如何工作	(330)
9.5.4	启动 OLE 拖动操作	(332)
9.5.5	将 OLE 拖动源拖到 OLE 拖放目标的上方	(334)
9.5.6	将 OLE 拖动源放到 OLE 拖放目标上	(337)
9.5.7	用鼠标和键盘修改放下效果和用户反馈	(339)
9.5.8	创建自定义数据格式	(340)
9.5.9	从“资源管理器”中拖动文件	(341)

9.6	自定义鼠标指针	(342)
9.6.1	图标和光标	(343)
9.7	响应键盘事件	(344)
9.7.1	编写低级键盘处理程序	(344)
9.7.2	KeyPress 事件	(345)
9.7.3	KeyDown 和 KeyUp 事件	(346)
9.7.4	编写窗体级键盘处理程序	(348)
第十章	文本与图形	(351)
10.1	使用字体	(351)
10.1.1	为应用程序选择字体	(351)
10.1.2	检查可用的字体	(351)
10.1.3	设置字体特征	(352)
10.1.4	把 Font 属性应用到特定的对象	(353)
10.1.5	FontTransparent 属性	(354)
10.2	在窗体和图片框上显示文本	(354)
10.2.1	使用 Print 方法	(354)
10.2.2	文本的截断	(354)
10.2.3	在指定位置显示 Print 输出	(355)
10.3	格式化数字、日期和时间	(357)
10.3.1	命名的格式	(358)
10.3.2	数字的格式	(359)
10.3.3	打印格式化的日期和时间	(359)
10.4	使用选定的文本	(359)
10.5	使用剪贴板对象传送文本和图形	(360)
10.5.1	使用剪贴板剪切、复制和粘贴文本	(360)
10.5.2	在剪贴板上使用多种格式	(362)
10.5.3	检查剪贴板上的数据格式	(364)
10.6	坐标系统概述	(364)
10.6.1	改变对象的坐标系统	(365)
10.6.2	刻度的转换	(368)
10.7	使用图形控件	(369)
10.7.1	给应用程序添加图片	(369)
10.7.2	窗体和控件的图形属性概述	(371)
10.7.3	用 AutoRedraw 创建持久的图形	(372)
10.7.4	用 ClipControls 裁剪区域	(373)
10.7.5	用 AutoRedraw 和 ClipControls 使图形分层	(374)
10.7.6	动态地移动控件	(376)
10.7.7	动态地改变控件大小	(379)

10.7.8	创建简单动画	(380)
10.8	使用图形方法	(380)
10.8.1	图形方法画图的原理	(381)
10.8.2	指定线宽	(382)
10.8.3	指定实线或虚线	(382)
10.8.4	DrawMode 的显示控制作用	(383)
10.8.5	窗体加载时创建图形	(384)
10.9	使用颜色	(385)
10.9.1	在运行时指定颜色	(385)
10.9.2	使用 RGB 函数	(385)
10.9.3	使用颜色属性	(385)
10.9.4	使用 256 种颜色	(387)
10.9.5	管理多个调色板	(388)
10.10	使用图片对象	(391)
10.10.1	使用图片对象数组	(391)
10.10.2	使用图片对象代替 Windows API	(392)
10.11	打印	(392)
10.11.1	从应用程序中打印	(392)
10.11.2	使用 Printer 对象	(393)
10.11.3	捕获打印机错误	(397)
第十一章	数据库操作	(399)
11.1	用 Data 控件访问数据库	(399)
11.1.1	Data 控件能做什么	(399)
11.1.2	Data 控件的属性	(400)
11.1.3	创建简单的数据库应用程序	(400)
11.1.4	数据库的设计和结构	(403)
11.1.5	记录集	(404)
11.2	使用数据绑定控件	(405)
11.2.1	绑定控件的类型	(406)
11.2.2	添加、更新和删除记录	(406)
11.2.3	使用 DBListBox 与 DBComboBox	(407)
11.2.4	使用 DBGrid	(410)
11.2.5	在 DBGrid 中使用范围	(413)
11.2.6	显示 DBGrid 中被计算的字段	(414)
11.3	用代码操作记录	(414)
11.3.1	记录集的定位	(414)
11.3.2	查找指定的记录	(416)
11.3.3	用代码更新记录集	(418)

11.3.4	控制事务.....	(420)
11.3.5	确认数据库的变化.....	(422)
11.3.6	数据访问专用的 Update 方法	(423)
11.4	处理文件.....	(424)
11.4.1	文件访问类型.....	(424)
11.4.2	文件访问函数和语句.....	(424)
11.4.3	使用顺序文件访问.....	(425)
11.4.4	使用随机文件访问.....	(427)
11.4.5	使用二进制文件访问.....	(429)
附录 A	Visual Basic 5.0 安装方法与 IDE 简介	(431)
A.1	版本简介	(431)
A.2	系统需求	(431)
A.3	安装	(431)
A.3.1	安装步骤	(432)
A.3.2	添加或删除 Visual Basic 5.0 部件	(432)
A.3.3	启动	(432)
A.3.4	获得帮助	(432)
A.4	集成开发环境	(433)
A.4.1	启动	(434)
A.4.2	集成开发环境的组成	(434)
A.4.3	环境选项	(435)

第一章 中文 Visual Basic 5.0 基础

本章将介绍 Visual Basic 应用程序的运行机制,并给出一个简单的 Visual Basic 应用程序示例。

1.1 Visual Basic 概念

为了理解应用程序开发过程,先要理解 Visual Basic 的一些关键概念。由于 Visual Basic 是 Windows 开发语言,所以它在某些方面与 Windows 环境有一定的相似性。

1. 交互式开发

Visual Basic 在编程者输入代码时便进行解释,即时捕获并突出显示大多数语法或拼写错误。除即时捕获错误以外,Visual Basic 也在输入代码时部分地编译该代码。当准备运行和测试应用程序时,只需极短时间即可完成编译。如果编译器发现了错误,则将错误突出显示于代码中。这时可以更正错误并继续编译,而不需从头开始。

由于 Visual Basic 的交互特性,因此在开发应用程序时,你自己正频繁地运行着自己的应用程序。通过这种方式,代码运行的效果可以在开发时进行测试,而不必等到编译完成以后。

2. 窗口、事件和消息

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。

Windows 操作系统通过给每个窗口指定一个唯一的标识号(窗口句柄或 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,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他窗口。然后,每一个窗口才能根据自身处理该条消息的指令而采取适当的操作(例如,当窗口解除了其他窗口的覆盖时,重显自身窗口)。

3. 属性、方法和事件

Visual Basic 的窗体和控件是具有自己的属性、方法和事件的对象。可以把属性看作一个对象的性质,把方法看作对象的动作,把事件看作对象的响应。

4. 事件驱动

在“过程化”的应用程序中,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。从第一行代码执行程序并按应用程序中预定的路径执行,必要时调用过程。

在事件驱动的应用程序中,代码不是按照预定的路径执行,而是在响应不同的事件时执行不同的代码片段。事件可以由用户操作触发、也可以由来自操作系统或其他应用程序的消息触发、甚至由应用程序本身的消息触发。这些事件的顺序决定了代码执行的顺序,因此应用程序每次运行时所经过的代码的路径都是不同的。

在执行,中代码也可以触发事件。例如,在程序中改变文本框中的文本将引发文本框的 Change 事件。如果 Change 事件中包含有代码,则将导致该代码的执行。如果原来假设该事件仅能由用户的交互操作所触发,则可能会产生意料之外的结果。正因为这一原因,所以在设计应用程序时理解事件驱动模型并牢记在心是非常重要的。

1.2 第一个 Visual Basic 应用程序

创建 Visual Basic 应用程序有四个主要步骤:创建应用程序界面,设置属性,编写代码,运行(测试)。为了说明这一实现过程,我们先按照以下步骤创建一个简单的应用程序,它由一个文本框和一个命令按钮组成。单击命令按钮,文本框中就会出现一条信息。

1.2.1 创建界面

窗体是创建应用程序的基础,通过使用窗体可将窗口和对话框添加到应用程序中,也可把窗体作为项的容器,这些项是应用程序界面中的不可视部分。例如,应用程序中可能有一个作为图形容器的窗体,而这些图形是打算在其他窗体中显示的。

建立 Visual Basic 应用程序的第一步是创建窗体,然后在创建的窗体上绘制构成界面的对象。对于本节的例子,可使用工具箱中的两个控件:

按钮

控件



文本框



命令按钮

1. 绘制控件

要用工具箱绘制控件:

- (1) 单击要绘制的控件的工具(此时是“文本框”)。
- (2) 将指针移到窗体上。该指针变成十字线,如图 1.1 所示。
- (3) 将十字线放在控件的左上角所在处。
- (4) 拖动十字线画出适合你要的控件大小的方框(拖动的意思是按住鼠标左键用鼠标指针移动对象)。
- (5) 释放鼠标按钮。控件出现在窗体上。

在窗体上添加控件的另一个简单方法是双击工具箱中的控件按钮。这样会在窗体中央创建一个尺寸为缺省值的控件;然后再将该控件移到窗体中的其他位置。