

Visual Basic 3.0 for Windows

程序设计问题大全

夏 雨 编著



西安交通大学出版社

Visual Basic 3.0 for Windows 程序设计问题大全

夏雨 编著

西安交通大学出版社

内容提要

本书以问题解答形式全面系统地介绍了 Visual Basic 3.0(专业版)for Windows,以及编程中可能遇到的各种问题。同时列举了大量实例并配有 180 余幅插图。

本书内容由浅入深,循序渐进,适合初学者使用。200 余个问题既相对独立,又相互联系;可以系统学习,也可以有针对性地阅读。

本书可作为学习 Visual Basic 的辅助读物,也可作为计算机用户的备查手册。

(陕)新登字 007 号

Visual Basic 3.0 for Windows

程序设计问题大全

夏雨 编著

责任编辑 胡瑞文 叶 涛

责任校对 仙静

西安交通大学出版社出版发行

(西安市咸宁西路 28 号 邮政编码 710049)

西安德力彩印厂印装

各地新华书店经销

*

开本:787×1092 1/16 印张:19.75 字数:480 千字

1996 年 11 月第 1 版 1996 年 11 月第 1 次印刷

印数:1——5000

ISBN7-5605-0837-5/TP·127 定价:20.00 元

若发现本社图书有倒页、白页、少页及影响阅读的质量问题,请去当地销售部门调换或与我社发行科联系调换。发行科电话:(029)3268357,3267874

前 言

自从 Windows 操作系统问世以来,由于它的图形用户界面(GUI),使计算机操作变得直观、容易和灵活,因而深受广大计算机用户的欢迎。现在,作为商品化的应用程序,它的用户界面几乎都是清一色的窗口图形界面。Microsoft 公司提供了一套开发窗口图形界面的工具——Microsoft Windows Software Development Kits(简称 SDK)。然而使用 SDK 来开发图形界面不是一般人所能为,需要熟悉 C 语言编程以及 SDK 软件包提供的 500 多个函数。有些函数的参数十分冗长,熟悉这些参数本身就是一件很烦人的事,这就使一些想编写 Windows 应用程序的人望而却步。有没有一种简单的设计工具让一般的计算机用户也能设计 Windows 的图形界面呢? Visual Basic 正是在这种需求下应运而生的。Visual Basic 是 Microsoft 公司于 1991 年推出的可视性编程工具,使用它能编写出 Windows 应用程序,设计出各种各样的窗口图形界面。使用 Visual Basic 不需要熟悉 C 语言和名目繁多的 SDK 软件包中的函数,只要你会使用 Windows,熟悉 Basic 语言,就会使用 Visual Basic。正是由于 Visual Basic 简单易学,因此它的出现立即受到人们的青睐,获得了广泛的应用。

Visual Basic 是一种面向对象的程序设计语言,既保持了 Windows 图形界面的可视性、操作直观简单等特点,又融进了 Basic 的编程能力。Visual Basic 与传统的编程语言不同。传统的编程语言是面向过程的,程序员必须按某种顺序逐个地完成过程,过程的执行是串行的。Visual Basic 是一种事件驱动的程序设计语言。事件是指按键或敲击鼠标按钮等动作,每当某个事件发生时,Visual Basic 就会执行相应的事件过程,完成指定的工作。因此事件过程是平行的。程序员不必考虑过程执行的次序,只要关心事件及事件过程就可以了,这样,应用程序的设计就变得简单灵活了。

Visual Basic 提供了许多可视性工具,利用这些工具加快了应用程序的开发。

1992 年秋天,Microsoft 公司推出了 Visual Basic 2.0 版。1993 年 Microsoft 公司又推出了 Visual Basic 3.0 版。3.0 版比 2.0 版作了许多改进,增加了十多个可视性设计工具,增加了许多事件、属性和方法,加强了程序调试能力,增加了数据库访问和三维图形外观功能。本书以 3.0 版为蓝本,全面介绍 Visual Basic 的各种功能。对于同类书籍中较少涉及的问题作了比较深入的研究,并把研究结果编入本书,以飨读者。

本书以问题解答形式呈现在读者面前,问题由浅入深,涉及到 Visual Basic 的方方面面。读者既可以循序渐进地学习,以期获得 Visual Basic 的全面知识,也可以有选择地学习某些问题解答,以期解决各自特有的问题。为阅读方便,本书按问题内容分为 20 章。第 1 章 Visual Basic 初步,第 2 章 Visual Basic 编程基础,第 3 章 Visual Basic 程序设计,第 4 章内部函数,第 5 章自定义函数和过程,第 6 章方法,第 7 章获取帮助信息,第 8 章窗体,第 9 章控制,第 10 章菜单设计,第 11 章图形绘制,第 12 章文件的输入和输出,第 13 章程序调试,第 14 章三维外观,第 15 章串行口通信,第 16 章剪贴板和 OLE2.0,第 17 章动态数据交换,第 18 章 Visual Basic 使用 DLL,第 19 章多媒体编程初步(MCI)和第 20 章访问数据库。

由于作者水平有限,书中错误在所难免,望读者批评指正。

编者 1995 年 9 月于上海

目 录

第 1 章 Visual Basic 初步

- 1.1 什么是 Visual Basic? (1)
- 1.2 运行 Visual Basic 3.0 需要什么样的软
硬件环境? (2)
- 1.3 如何安装 Visual Basic 3.0? (2)
- 1.4 如何启动 Visual Basic 3.0? (2)
- 1.5 Visual Basic 有哪些基本术语? (2)
- 1.6 常用的鼠标和键盘操作有哪些? (4)
- 1.7 Visual Basic 的屏幕由哪些部分组成?
..... (5)
- 1.8 怎样创建一个应用程序? (9)
- 1.9 如何运行应用程序? (12)
- 1.10 怎样保存应用程序? (12)
- 1.11 如何生成可执行文件? (13)

第 2 章 Visual Basic 编程基础

- 2.1 Visual Basic 中有哪些数据类型?
..... (15)
- 2.2 Visual Basic 中如何表示常量? (17)
- 2.3 Visual Basic 如何表示变量? (17)
- 2.4 变量有哪些类型? (18)
- 2.5 什么是变量的作用域? (24)
- 2.6 什么是局部变量? (24)
- 2.7 什么是窗体变量? (25)
- 2.8 什么是模块变量? (26)
- 2.9 什么是全局变量? (27)
- 2.10 如何获得 Variant 的当前数据类型?
..... (28)
- 2.11 Visual Basic 有哪些运算符? (29)
- 2.12 运算符的优先级是如何排列的?
..... (35)
- 2.13 表达式及其运算结果的数据类型
如何确定? (36)

第 3 章 Visual Basic 程序设计

- 3.1 Visual Basic 有哪些基本语句? (37)
- 3.2 如何书写赋值语句? 赋值语句有什
么功能? (38)
- 3.3 什么是注释语句? (39)
- 3.4 什么是条件分支语句? (39)
- 3.5 如何使用 If Then Else 语句? (39)
- 3.6 如何使用 Select Case 语句? (41)
- 3.7 为什么要使用循环语句? (42)
- 3.8 如何使用 For...Next 语句? (42)
- 3.9 如何使用 Do...Loop 语句? (44)
- 3.10 如何使用 While...Wend 语句? (45)
- 3.11 什么是转移语句? (45)
- 3.12 如何使用 Go To 语句? (45)
- 3.13 如何使用 On...GoSub 和 On...GoTo
语句? (46)
- 3.14 如何使用 GoSub...Return 语句?
..... (48)
- 3.15 如何使用 On Error GoTo 语句?
..... (49)

第 4 章 内部函数

- 4.1 Visual Basic 提供哪些内部函数? ... (50)
- 4.2 Visual Basic 提供哪些数学函数? ... (50)
- 4.3 Visual Basic 提供哪些字符串函数?
..... (53)
- 4.4 Visual Basic 提供哪些日期型函数?
..... (57)
- 4.5 Visual Basic 提供哪些转换函数? ... (61)
- 4.6 Visual Basic 提供哪些输入输出函数?
..... (63)
- 4.7 Visual Basic 提供哪些测试函数? ... (71)
- 4.8 什么是 Shell 函数? (73)
- 4.9 如何使用 Do Events 函数? (74)

第5章 自定义函数和过程

- 5.1 如何建立自定义函数? (76)
- 5.2 什么是过程? (80)
- 5.3 如何建立过程? (80)
- 5.4 如何在过程中传递参数? (81)
- 5.5 如何调用过程? (81)

第6章 方法

- 6.1 什么是方法? (85)
- 6.2 如何使用 Print 方法? (85)
- 6.3 如何使用 Cls 方法? (88)
- 6.4 如何使用 Move 方法? (88)
- 6.5 如何使用 Scale 方法? (89)
- 6.6 如何使用 TextHeight 和 TextWidth 方法? (90)
- 6.7 如何使用 Show 方法? (92)
- 6.8 如何使用 Hide 方法? (92)

第7章 获取帮助信息

- 7.1 如何打开帮助菜单? (93)
- 7.2 如何访问帮助主题? (94)
- 7.3 如何使用主题窗口的菜单? (97)
- 7.4 如何搜索主题? (98)

第8章 窗体

- 8.1 什么是窗体? (99)
- 8.2 窗体有哪些属性? (99)
- 8.3 如何定义窗体坐标系? (106)
- 8.4 什么是 MDI 窗体? (108)
- 8.5 如何建立 MDI 窗体? (108)
- 8.6 MDI 窗体有哪些属性? (109)
- 8.7 MDI 窗体使用哪些方法? (109)
- 8.8 如何使用窗体? (109)
- 8.9 窗体有哪些事件? (111)
- 8.10 如何打印窗体属性和代码? (118)

第9章 控制

- 9.1 什么是标签控制? (119)
- 9.2 标签控制有哪些主要属性? (119)
- 9.3 标签控制可发生哪些事件? (122)
- 9.4 标签控制使用哪些方法? (124)
- 9.5 什么是文本框控制? (124)
- 9.6 文本框控制有哪些属性? (124)
- 9.7 文本框控制可发生哪些事件? (126)
- 9.8 文本框控制可使用哪些方法? (126)
- 9.9 如何使用文本框控制? (126)
- 9.10 什么是命令按钮控制? (128)
- 9.11 命令按钮控制有哪些属性? (128)
- 9.12 命令按钮控制可发生哪些事件? (129)
- 9.13 命令按钮控制可使用哪些方法? (129)
- 9.14 如何使用命令按钮控制? (131)
- 9.15 什么是选择按钮和检查框控制? (133)
- 9.16 选择按钮和检查框控制有哪些属性? (133)
- 9.17 选择按钮和检查框控制可发生哪些事件? (133)
- 9.18 选择按钮和检查框控制可使用哪些方法? (133)
- 9.19 如何使用选择按钮和检查框控制? (133)
- 9.20 什么是列表框控制? (135)
- 9.21 列表框控制有哪些属性? (135)
- 9.22 列表框控制可发生哪些事件? (137)
- 9.23 列表框控制可使用哪些方法? (137)
- 9.24 如何使用列表框控制? (138)
- 9.25 什么是组合框控制? (140)
- 9.26 组合框控制有哪些属性? (140)
- 9.27 组合框控制可发生哪些事件? (141)
- 9.28 组合框控制可使用哪些方法? (141)
- 9.29 如何使用组合框控制? (141)

9.30 什么是滚动条控制?	(144)	9.38 框架控制可使用哪些方法?	(147)
9.31 滚动条控制有哪些属性?	(144)	9.39 如何使用框架控制?	(147)
9.32 滚动条控制可发生哪些事件?	(145)	9.40 什么是计时器控制?	(148)
9.33 滚动条控制可使用哪些方法?	(145)	9.41 计时器控制有哪些属性?	(148)
9.34 如何使用滚动条控制?	(145)	9.42 计时器控制可发生哪些事件? ...	(149)
9.35 什么是框架控制?	(146)	9.43 计时器控制可使用哪些方法? ...	(149)
9.36 框架控制有哪些属性?	(146)	9.44 如何使用计时器控制?	(149)
9.37 框架控制可发生哪些事件?	(147)		

第 10 章 菜单设计

10.1 如何建立菜单?	(151)	10.3 如何创建弹出式菜单?	(157)
10.2 菜单有哪些属性?	(156)		

第 11 章 图形绘制

11.1 如何在程序中载入图形?	(159)		(166)
11.2 如何认识 VB 的坐标系统?	(161)	11.7 如何在绘图中使用颜色?	(168)
11.3 如何绘制直线?	(162)	11.8 如何正确使用图片框和图象控制?	
11.4 如何绘制点?	(164)		(170)
11.5 如何绘制圆?	(165)	11.9 如何使用窗体的图标显示?	(173)
11.6 如何使用 Shape(几何图形)控制?		11.10 如何实现图形的拖放?	(175)

第 12 章 文件的输入和输出

12.1 如何使用驱动器、目录和文件三个列表框?	(181)	12.3 如何打开和关闭文件?	(188)
12.2 如何使用通用文件控制对话框?	(184)	12.4 如何写文件?	(190)
		12.5 如何读文件?	(191)
		12.6 如何在 VB 中使用打印机?	(193)

第 13 章 程序调试

13.1 什么是 Visual Basic 的调试环境?	(195)	13.4 如何使用断点(Breakpoint)?	(200)
13.2 如何使用 Debug 窗口?	(196)	13.5 如何观察程序的执行顺序和相互调用关系?	(201)
13.3 如何使用观察窗口?	(199)		

第 14 章 三维外观

14.1 为什么要使用三维外观?	(204)	14.5 如何使用三维按钮组?	(210)
14.2 如何使用三维面板?	(205)	14.6 如何使用三维检查框和三维选择按钮?	(212)
14.3 如何使用三维框架?	(208)	14.7 如何使用动画按钮?	(213)
14.4 如何使用三维命令按钮?	(209)		

第 15 章 串行口通信

15.1 何谓串行口通信?	(217)	15.3 如何使用 MSCOMM. VBX 实现通信?	(219)
15.2 如何实现电话拨号功能?	(218)		

15.4 如何更进一步使用 MSCOMM.VBX?	(224)
---------------------------------	-------

第 16 章 剪贴板和 OLE2.0

16.1 剪贴板的原理是什么?	(228)	16.5 如何使用 VB 的 OLE2 控制来设计窗体?	(233)
16.2 如何选择文本?	(229)	16.6 如何使用 OLE 属性?	(234)
16.3 如何操作剪贴板上的数据?	(230)	16.7 如何操作其它 Windows 应用程序或窗体?	(239)
16.4 什么是对象的链接和嵌入?	(232)		

第 17 章 动态数据交换

17.1 什么是动态数据交换?	(242)	17.5 如何使用 DDE 方法(Method)?	(247)
17.2 如何在窗体设计阶段建立 DDE?	(243)	17.6 如何建立一个简单的 DDE 程序?	(249)
17.3 如何使用 DDE 属性?	(244)		
17.4 如何使用 DDE 事件?	(246)		

第 18 章 Visual Basic 使用 DLL

18.1 什么是 DLL?	(254)	18.3 如何在程序中调用 DLL 子程序?	(256)
18.2 如何对 DLL 子程序进行声明?	(254)		

第 19 章 多媒体编程初步

19.1 什么是多媒体环境?	(267)	19.5 什么是媒体播放器?	(277)
19.2 什么是多媒体 Windows MCI? ...	(268)	19.6 如何制作一个 CD 播放器?	(281)
19.3 如何使用 MCI 控制的属性?	(271)	19.7 如何制作声音播放器?	(285)
19.4 如何使用 MCI 控制的事件?	(275)		

第 20 章 访问数据库

20.1 什么是 Access 关系式数据库? ...	(288)	20.6 如何在数据库中查找记录?	(296)
20.2 BIBLIO.DMB 是一个怎样的数据库?	(289)	20.7 什么是网格控制?	(296)
20.3 怎样创建 Data 控制?	(290)	20.8 网格控制有哪些属性?	(297)
20.4 如何访问数据库?	(290)	20.9 网格控制有哪些事件?	(301)
20.5 如何显示数据库的内容?	(293)	20.10 如何在网格中浏览数据库?	(303)
		20.11 如何进行数据库查询?	(306)

第 1 章 Visual Basic 初步

本章介绍什么是 Visual Basic;运行 Visual Basic 需要什么样的环境;怎样启动 Visual Basic 以及如何创建一个应用程序。学习本章之后,你将对怎样在 Visual Basic 环境中设计 Windows 应用程序有一个基本了解,开始进入 Visual Basic 大门。

1.1 什么是 Visual Basic?

Visual Basic 中的 Visual 是“可视”的意思。因此 Visual Basic 是一种可视性程序设计语言。它与传统的程序设计语言不同,区别在于:传统的程序设计语言是面向过程的,按顺序执行每个过程,因而传统的程序设计是串行的、呆板的;而 Visual Basic 是面向对象的、可视的、并由事件驱动的程序设计语言,因而 Visual Basic 是并行的、生动的。事件是指发生某个动作,例如敲击鼠标器和按键等。在 Visual Basic 中每个事件对应一个过程,当事件发生时,对应的过程便运行,完成指定的工作。这些事件的驱动并没有次序的先后,因此事件的驱动是并行的。Visual Basic 提供了许多可视性设计工具,利用这些工具可以很方便地设计出窗口和菜单等。如果使用传统的程序设计语言设计一个下拉式菜单可能需要编写十几页到几十页的程序代码,但使用 Visual Basic 只需利用菜单工具加十几行程序代码即可。Visual Basic 的可视性更主要地表现在应用程序设计的图形化界面上。整个应用程序是一个窗口和图形组成的界面,直观、清晰。对每个窗口和图形可单独调试,所有输入和输出都在窗口上进行,使程序设计变得简单和容易。

综上所述,Visual Basic 既保留了 Windows 图形界面的特点,又融进了 Basic 编程的能力,使 Windows 应用程序设计对大多数程序员来说不再是一种奢望,设计一个具有类似 Windows 图形界面的应用程序变得简单和容易了。

Visual Basic 最早是由 Microsoft 公司在 1991 年推出的。1992 年秋天,Microsoft 公司又推出了 Visual Basic 2.0 版,该版本一经推出,便获得了广泛的应用。1993 年 Microsoft 又开发了 Visual Basic 3.0 版。3.0 版与 2.0 版比较,结构化和功能均有了很大提高。主要体现在下列几个方面:

- ① 增加了许多新的设计工具。
- ② 增加了许多事件、属性和方法。
- ③ 增加了数据库访问功能。
- ④ 增加了三维图形外观功能。
- ⑤ 增加了程序调试能力。

本书下面几章将全面介绍 Visual Basic 3.0 的各种功能及程序设计方法。

1.2 运行 Visual Basic 3.0 需要什么样的软硬件环境?

Visual Basic 3.0 可在 Windows 3.0 版以上环境下运行。既可在西文 Windows 环境下运行,也可以在中文 Windows 环境下运行。它通常要求 Windows 以 386 增强方式运行。因此需要下列硬件的支持:

- 80386 或更高一档微机;
- 至少 2MB 以上内存(推荐 4MB 以上内存);
- 40MB 以上硬盘;
- 1.2MB 或 1.44MB 软盘驱动器;
- 支持 Windows 3.0 的视频显示器(VGA 等);
- 鼠标器(任选)。

1.3 如何安装 Visual Basic 3.0?

Visual Basic 3.0 的安装与 Windows 的安装非常类似。具体步骤如下:

- (1) 首先启动 MS-DOS 或 PC-DOS(3.1 以上版本)。
- (2) 启动 Windows。先进入 Windows 子目录,然后键入 Win 命令。
- (3) 在软盘驱动器中插入安装盘(DISK1#盘)。
- (4) 在 Windows 的程序管理器(Program Manager)窗口中,打开文件(File)菜单,选择运行(Run)命令。此时会出现一个对话框,提示你输入要运行的文件名。你可以键入 A:SETUP.EXE 或 B:SETUP.EXE(视软盘驱动器的设定而定)。
- (5) 遵循屏幕上的提示,输入各任选项,插入后续软盘完成安装。安装完成后,SETUP.EXE 会自动把 Visual Basic 3.0 作为一个程序组出现在程序管理器(Program Manager)窗口上。

1.4 如何启动 Visual Basic 3.0?

启动 Visual Basic 有两种方法:第一种方法是先启动 Windows,然后在 Windows 的程序管理器(Program Manager)窗口中,双击“Visual Basic 3.0”图标(Icon)。双击是指把鼠标器指针移动到“Visual Basic 3.0”图标上,迅速连击两次鼠标器按钮(通常是左按钮)。第二种方法是在 DOS 环境的命令提示行中直接键入 Win VB\VB,同时启动 Windows 和 Visual Basic 3.0。

1.5 Visual Basic 有哪些基本术语?

Visual Basic 有一些基本术语,弄清楚这些基本术语的含义对阅读后面的问题会有很大帮助。基本术语包括项目(Project)、窗体(Form)、控制(Control)、对象(Object)、属性(Properties)、事件(Event)、事件过程(Event Procedure)、方法(Method)和项(Item)等。下面介绍这些术语。

- (1) 项目(project)

项目是窗体(Form)和控制(Control)的集合。一个项目对应一个应用程序,至少包含一个窗体。通常一个项目包含一个或多个窗体、若干个控制以及模块(Module)。项目、窗体和模块都以文件形式存储在磁盘中,分别称为项目文件、窗体文件和模块文件。项目文件的扩展名是 MAK;窗体文件的扩展名是 FRM;模块文件的扩展名是 BAS。图 1.1 描述了项目的组成。

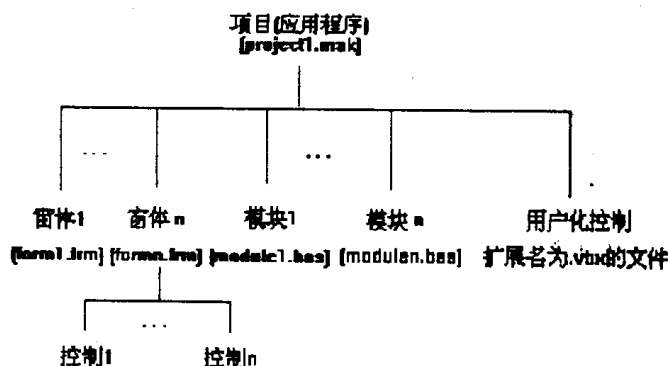


图 1.1

扩展名为 VBX 的文件称为用户化控制(custom control),它们可根据用户的需求加入到项目中或从项目中去掉。每当你新建一个项目时,Visual Basic 会自动把一些基本的用户化控制加到项目中去。每一个 VBX 文件对应工具箱(Tool Box)中的一个工具。每个窗体中的控制属于该窗体,它们与窗体一起作为项目文件被保存起来。

(2) 窗体(Form)

窗体是一个窗口,它包含若干个控制(control)和与这些控制对应的代码(code)。窗体是应用程序的界面,每个应用程序至少包含一个窗体。窗体用来显示控制、图形及其他窗体。

(3) 控制(control)

控制是窗体中的子窗口,它完成指定的输入和输出。它在窗体中表现为一个图形。例如,为了在应用程序中输入数据,你应该创建一个文本框(Text Box)控制,应用程序将从该控制中接收数据。为了输出应用程序的结果,你可以创建一个标签框(Label Box)或文本框控制,该控制用来显示结果。控制的种类很多,各种控制的功能将在后续的章节中逐步介绍。

(4) 对象(object)

对象是窗体、控制和设备等的总称。窗体和控制是 Visual Basic 中使用最频繁的对象。设备也作为对象,例如打印机(printer)和显示屏幕(Screen)都是对象。

(5) 属性(properties)

属性是对象所具有的特性。一个对象有一组对应的属性集。例如,窗体有标题(Caption)、名称(Name)等属性。不同的对象其属性集是不同的。每一个属性有一个属性值。每当你创建一个新的窗体或控制时,Visual Basic 会自动为该窗体或控制的属性设置缺省的属性值。你可以通过重新设置属性的值来改变这些属性。属性值的设置有如下格式:

对象名.属性 = 属性值

例如,设有一个文本框控制,其控制名(Name)Text1,于是

Text1.text = "Example"

把字符串"Example"赋给 Text1 的 text 属性。

(6) 事件(Event)

事件是指 Visual Basic 所能识别的动作。例如,按键(Keypress)动作是一个事件,敲击鼠标按钮(Click)也是一个事件。不同的对象所能识别的事件是不相同的。每当你创建一窗体或控制时,Visual Basic 会自动列出窗体或控制识别的事件。

(7) 事件过程(Event Procedure)

事件过程是一组代码(Code),它受事件驱动完成某个指定的功能。事件过程的一般格式如下:

```
Sub 对象名_事件名()  
    ... (程序代码)  
End Sub
```

对象名是指窗体名或控制名(Name),事件名是事件的名称,通常可通过事件列表选择。

(8) 方法(Method)

方法是一种特殊的过程和函数。Visual Basic 为每个对象包含了方法。方法是特定对象的一部分,使用方法时必须指明对象。它有如下格式:

对象名.方法

例如,Print 是一个方法,它类似于 C 语言中的 printf 函数,用来在对象上显示信息。于是 text1.print "How are you!"

就在 Text1 控制上显示"How are you!"。

应该注意的是,有些方法属于所有对象,有些方法只属于少数对象,在使用之前应查明方法可应用的对象。

(9) 项(Item)

项是菜单中的一个元素。例如,文件(File)菜单中的 Save 和 Save as 等都是项。菜单中的项也称菜单项。通常一个项对应一条命令,所以选择一个菜单项也可以说是执行一条命令。

注意,不要把项(Item)和项目(Project)混淆起来。

1.6 常用的鼠标和键盘操作有哪些?

(1) 单击(Click)

单击是指用手指敲击鼠标(Mouse)按钮(通常为左按钮)一次,然后松开。

(2) 双击(Double Click)

双击是指连续两次快速敲击鼠标按钮(通常为左按钮),然后松开。

(3) 拖放(Drag Drop)

拖放是指按住鼠标按钮不放,移动鼠标指针(指针所在位置的图形也一起移动)至新位置,然后松开。

(4) 组合键

组合键用 ALT + F4 的形式表示。ALT + F4 表示同时按下 ALT 和 F4 键。

1.7 Visual Basic 的屏幕由哪些部分组成?

Visual Basic 初次启动后,屏幕上会出现 5 个窗口,如图 1.2 所示。现在分别介绍这 5 个窗口。

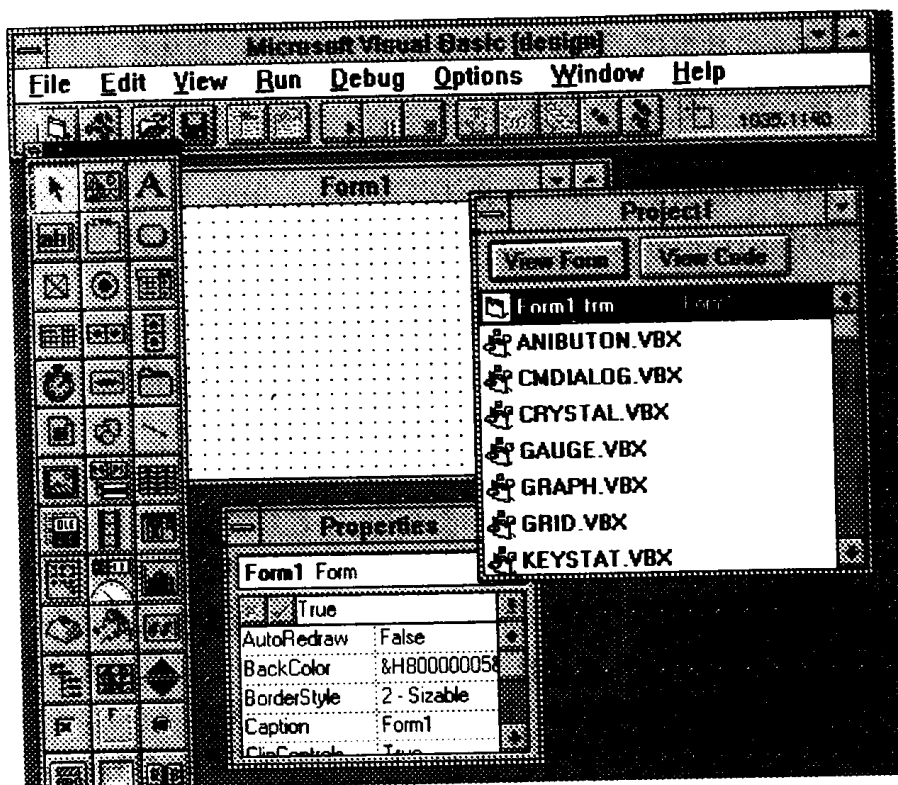


图 1.2

(1) 主窗口

最上面的一个窗口是主窗口。主窗口中有 3 个狭长条。第一个狭长条是 Visual Basic 的标题,显示 Microsoft Visual Basic[design]。其中 design 表示当前处于设计状态(Visual Basic 有两个状态:设计状态及运行状态。两个状态下的屏幕显示是不同的),中间的狭长条是菜单条,其上有 8 个菜单,分别是 File(文件)、Edit(编辑)、Run(运行)、Debug(程序调试)、Options(任选项)、Windows(窗口)和 Help(帮助)。最下边的一个狭长条是工具条,它提供了 File 菜单的大部分功能。设计工具条的目的是为了使用方便,用户可直接单击工具条中的工具而不必打开 File 菜单可达到同样的目的。工具条中的每一个工具的功能如图 1.3 所示。

在工具条的右边有两个方框,分别显示当前活动对象屏幕中的位置和大小。数字的单位是 twip。1 440twips $\approx$ 2.54cm(1 英寸),567twips $\approx$ 1cm,在启动 Visual Basic 时,窗体是当前活动对象,因此方框中显示的是窗体在屏幕中的位置和尺寸。

(2) 窗体窗口

主窗口下面的中间部分是窗体窗口(Form1)。Form1 是窗体的缺省标题,允许你通过下面将要介绍的窗口属性改变窗体的标题。每当你新建一个项目(Project)时,Visual Basic 总是为



图 1.3

你新建一个标题为 Form1 的窗口。相应的文件名为 Form1.frm。窗体窗口是窗体(Form)类对象,它是应用程序的界面。在这个界面上你可以设计各种各样的控制(control),以满足应用程序的需要。

窗体窗口的内容可以存储起来,缺省的文件名为 Form1.frm。你可以通过文件(File)菜单上的 Save 命令保存该文件,或通过文件(File)菜单上的 Save as 命令以另一个文件名保存该文件。窗体文件的扩展名为 FRM。

(3) 工具箱

在屏幕左侧有一个工具箱(Tool Box),其中每一个方框是一个工具,每一个工具对应一种控制。例如,标有字母 ab 的方框是一个文本框控制,用它可以在窗体上设计文本框。文本框主要用于文本输入和输出。

工具箱为窗体设计提供了极大的方便,你可以使用工具箱设计出各种各样的应用程序的用户界面。Visual Basic3.0 的工具箱比 2.0 版又增加了许多工具。每当你新增一个项目时,Visual Basic 自动为你配置常用的工具箱。工具箱中的工具可以根据需要增加和删除。工具箱中的每一个工具对应一个扩展名为 VBX 的文件。当某个文件从项目中被删除时,该文件对应的工具也从工具箱中被删除。删除工具箱中的工具的具体过程如下:

- ① 在项目窗口中选择一个扩展名为 VBX 的文件(项目窗口稍后介绍)。
- ② 打开 File 菜单,选择 Remove File 命令。

在工具箱中增加工具的过程如下:

- ① 打开 File 菜单,选择 Add File 命令,此时会出现图 1.4 的对话框。

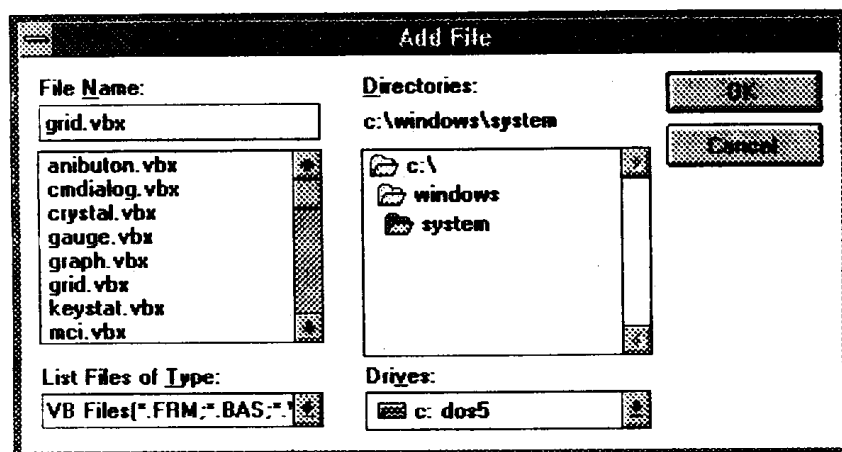


图 1.4

② 在对话框中输入某个扩展名为 VBX 的文件。

③ 按 OK 按钮,于是在工具箱中会出现与所加 VBX 文件对应的工具图标。

在设计状态下,工具箱通常总是出现的。如果你想隐去工具箱,以便使其他窗口有更多的空间,则可单击工具箱左上角的控制菜单框,选择“关闭”命令。若要再次显示工具箱,可单击 Windows 菜单,选择 Took Box 菜单项。

在运行状态下,工具箱自动隐去。图 1.5 标出了工具箱中各个工具图标的名称。这些工具的使用将在第 8.9 章窗体与控制的有关问题中详细介绍。

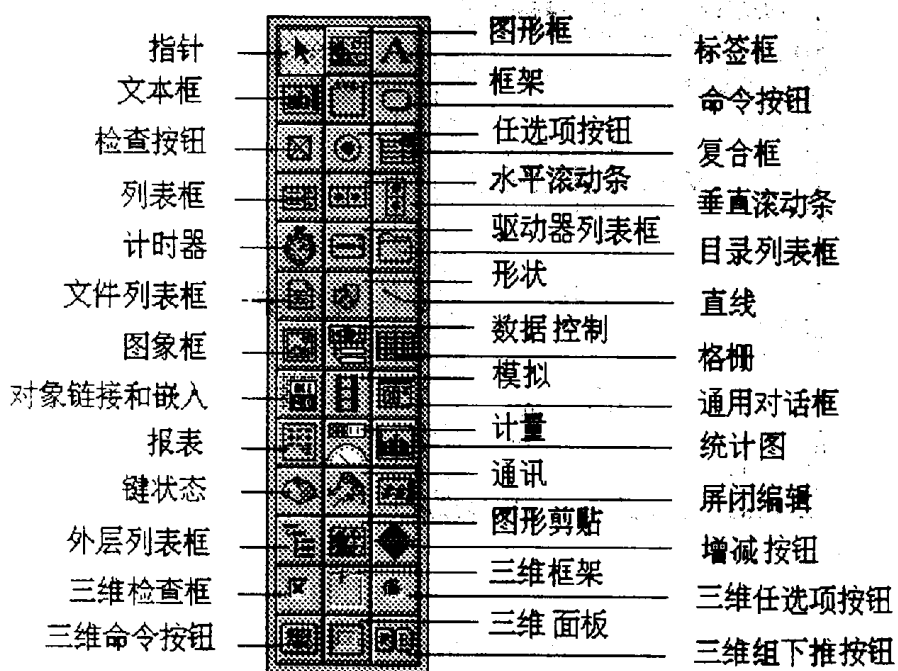


图 1.5

(4) 项目窗口

标题为 Project1 的窗口是项目窗口。项目窗口列出了应用程序所需要的全部文件清单。项目窗口至少包含一个窗体窗口,缺省的窗体窗口是 Form1。一个项目可以包含多个窗体,每个窗体对应一个扩展名为 FRM 的文件。每当新建一个项目时,新建项目总是包含一个文件名为 Form1.frm 的缺省的窗体窗口。项目窗口中的扩展名 VBX 文件是一些用户化控制,每一个用户化控制对应工具箱中的一个图标。你可以根据应用程序的需要,在项目中增加和删除 VBX 文件。每当你在项目中增加和删除 VBX 文件时,工具箱中的工具图标也会随之变化。增加和删除 VBX 文件的方法见前面关于工具箱的叙述。

项目窗口中还可以包含一些模块文件(扩展名为 BAS 的文件)。模块文件用来声明项目范围内使用的全局变量或编写项目使用的过程代码(Code)。代码是用 Visual Basic 编写的程序,该程序可被项目中各窗体调用。

项目窗口中还有两个命令按钮。一个是 View form,用来显示窗体窗口;另一个是 View Code,用来显示代码。若要显示窗体窗口,用鼠标单击 View Form 按钮;若要显示代码窗口,用鼠标单击 View Code 按钮。利用这两个按钮,你可以方便地修改窗体设计和代码。

有时,项目窗口会被窗体窗口遮住,此时你可以打开 Windows 菜单,用鼠标单击 Project 菜单项,使 Project 窗口可见。通过项目窗口,你可以改变当前活动的对象。例如,你可双击项目窗口中的 Form1.frm 项,使窗体 Form1 变成活动对象。你也可双击某个 BAS 文件,使该模块成为当前活动的对象。

(5) 属性窗口

标题为 Properties 的窗口是一个属性窗口。属性窗口由三部分组成:对象框、设置框和属性列表(见图 1.6)。

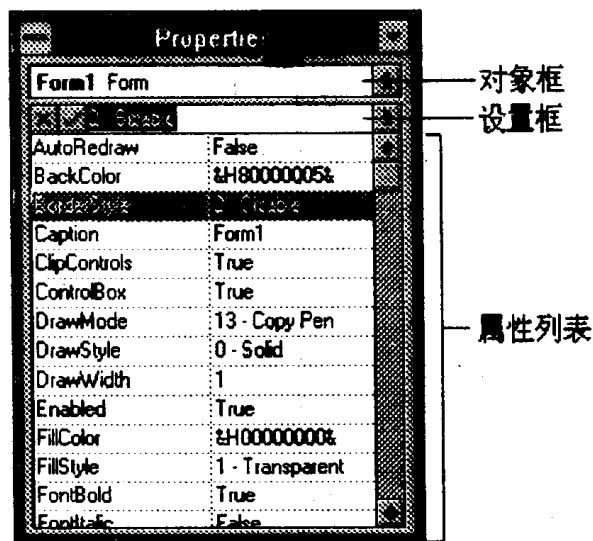


图 1.6

每一个对象(窗体或控制)有一个属性窗口,如果一个项目中包含了多个对象,你可以用鼠标单击对象框中的箭头,列出所有对象,然后选择一个对象,作为要设置属性的对象。

设置框中显示的是某个属性的缺省值。例如,对于 Form1 窗体,如果正在设置 Caption 属性,则设置框显示 Form1,表示窗体 Form1 的标题有缺省值 Form1。如果此时,你在设置框输入另一个字符串,然后用鼠标单击设置框中的带✓的按钮,则该字符串就成了窗体 Form1 的属性。如果你不想使输入的字符串成为窗体 Form1 的标题,可用鼠标单击设置框中带×的按钮,于是窗体 Form1 仍使用原来的标题。

有些属性以列表形式给出属性值。例如,Borderstyle 属性描述对象的边界取何种风格,它以列表形式给出,如图 1.7 所示。有些属性的设置框右边方框中有 3 个圆点,表示这些属性与图形、图象或图标有关。例如 Fill Color 属性设置图形的填充颜色,单击有 3 个圆点的方框时会给出一幅调色板,你可从中选择一种颜色作为属性值(见图 1.8)。

有些属性以值的形式给出,此时需要在设置框中输入新的值,再单击✓按钮,才能改变属性值。属性值也可以通过编写代码给出。例如

```
Form1.caption = "MyForm"
```

使窗体 Form1 的属性 Caption 有值 My Form。

属性列表中列出了对象的全部属性,你可以通过右边的垂直滚动条,浏览整个列表。

如果你想显示或改变某个属性的值,只要用鼠标单击该属性,于是该属性原来的值就显示

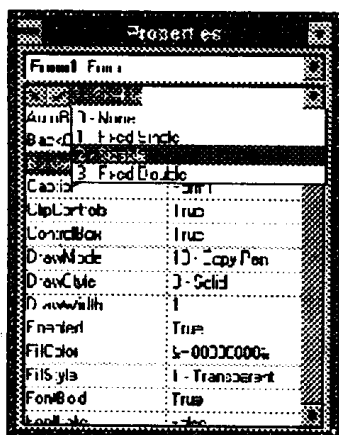


图 1.7

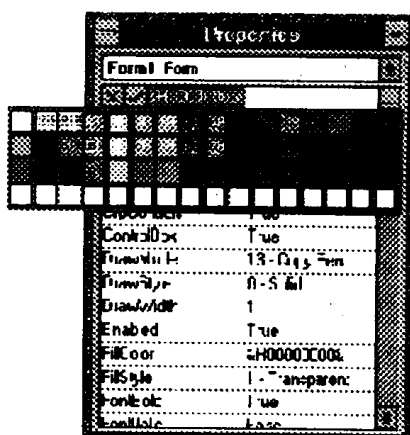


图 1.8

在设置框中。

属性列表有可能被窗体遮住,此时你可以用鼠标单击 Windows 菜单,然后选择 Properties 菜单项,使属性窗口可见。

1.8 怎样创建一个应用程序?

假设要创建一个求一个数的平方的应用程序,即输入一个数,输出这个数的平方。这个应用程序由 1 个窗体组成,窗体中有 4 个控制: 2 个标签框和 2 个文本框。完成后的窗体应有图 1.9 的形状。

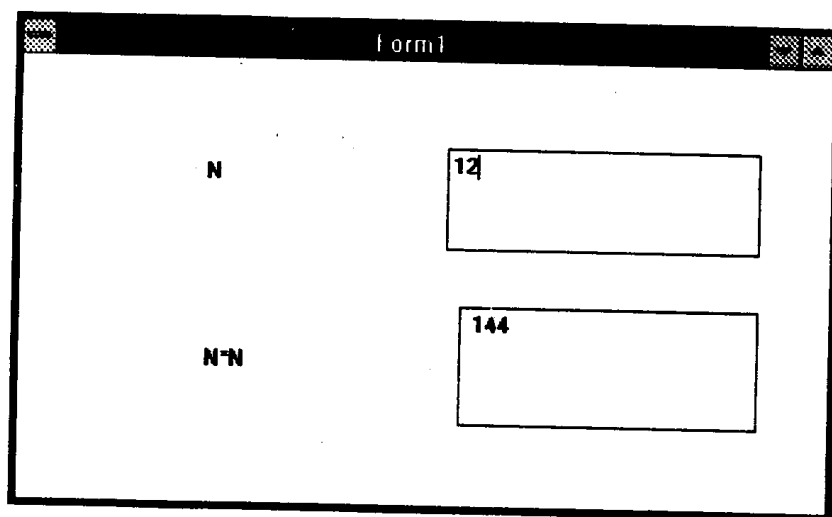


图 1.9

建立一个应用程序通常需要经过三个步骤:建立一个界面(窗体);设置窗体和控制属性;编写代码(程序)。

(1) 建立一个界面