

计算机软件开发系列丛书

学苑出版社

VISUAL BASIC

应用开发原理和技术

李春葆
章启俊 编著
王广泰



希望

计算机软件开发系列丛书

Visual Basic 应用开发原理和技术

李春葆
章启俊 编著
王广泰
燕卫华 审校

学苑出版社

1994

(京)新登字 151 号

内 容 提 要

本书全面讨论了 Visual Basic 的语言环境和应用开发技术,全书分为两个主要部分,第一章到第十二章介绍了 Visual Basic 的基本内容,后面几章讨论应用设计技术,为用户的应用开发提供了一个模板。本书内容详尽,结构严谨,可作为广大计算机编程人员的参考资料及各大专院校的教材。

欲购本书的用户,请直接与北京海淀 8721 信箱希望公司书刊部联系,邮政编码:100080,电话:2562329。

计算机软件开发系列丛书
Visual Basic 应用开发原理和技术

编 著:李春葆 章启俊 王广泰
审 校:燕卫华
责任编辑:甄国宪
出版发行:学苑出版社 邮政编码:100036
社 址:北京市海淀区万寿路西街 11 号
印 刷:北京东升印刷厂
开 本:787×1092 1/16
印 张:18.625 字 数:430 千字
印 数:1~5000 册
版 次:1994 年 7 月北京第 1 版第 1 次
ISBN7-5077-0779-2/TP·11
本册定价:19.00 元

学苑版图书印、装错误可随时退换

前 言

Microsoft Windows 是目前广泛使用的桌面操作系统,它带有强大的图形用户界面,具有灵活的窗口操作功能和一系列的应用工具。目前用 C 语言编程是 Windows 下编制程序的一条常规途径,但这不是一件轻松的工作,因为用户不得不学习 Windows 的输入输出以及 C 的精确语法,另一条在 Windows 下编程的途径便是使用一种可视的编程语言,Visual Basic 是这一类语言。

Visual Basic 提供了一套全新的编程工具和环境,一旦建立了用户所能看到的屏幕图像,就很容易设计与屏幕进行信息交换的代码,其先进的屏幕设计给用户提供了一种建立串行驱动的 Windows 应用程序的结构,在 Visual Basic 中书写代码很简单,仅仅需要在窗体编辑器中两次激活任一控制对象并把子例程的骨架调入文本编辑器。Visual Basic 既是一种新的宏语言,又是应用软件的高度集成,它提供了编译、项目管理和图形操作等功能。

本书全面讨论了 Visual Basic 的语言环境和应用开发技术,全书分为两个主要部分,第一章到第十二章介绍了 Visual Basic 的基本内容,后面几章讨论应用设计技术,为用户的应用开发提供了一个模板。

第一章介绍了 Visual Basic 的一般知识,第二章描述了用户界面的使用方法;第三章讨论了如何建立应用的界面;第四章讨论更高级的应用界面设计;第五章讲述编写代码的方法;第六章讨论 Visual Basic 作为语言的一些基本知识;第七章讨论各种函数,第八章描述命令钮和菜单的设计过程;第九章讲述信息输入窗体的设计技术;第十章讲述信息输出窗体的设计方法;第十一章讨论基本的图形设计技术。

第十二章讨论几个 Visual Basic 设计数值应用的例子;第十三章讨论高级图像、图形程序设计;第十四章讨论一些获取系统资源信息的例子;第十五章讨论几个金融应用的例子;第十六章讨论数据库应用设计;第十七章讨论绘制一些数学方程图形的例子,第十八章讨论绘制表示图的应用。由于编者水平有限,错误和不是之处难免,恳请读者批评指正。

本书的出版得到秦人华高级工程师的大力支持,编者在此表示衷心感谢。

编者

94.6

目 录

第一章 Visual Basic 概述	(1)
1.1 什么是 Visual Basic	(1)
1.2 安装和运行 Visual Basic	(2)
1.3 标准控制	(6)
1.4 控制属性	(8)
1.5 对象	(11)
1.6 实例设计	(13)
1.7 Visual Basic 能设计什么样的应用	(15)
第二章 Visual Basic 用户界面	(17)
2.1 设计窗口	(17)
2.2 窗体窗口	(22)
2.3 工具箱	(22)
2.4 项目窗口	(22)
第三章 建立界面	(23)
3.1 控制的操作	(23)
3.2 实例设计	(26)
3.3 高级窗体控制	(28)
3.4 建立界面的规则	(33)
第四章 设置界面属性	(34)
4.1 属性	(34)
4.2 设置属性	(41)
4.3 设置颜色	(48)
4.4 实例设计	(49)
第五章 编写代码	(51)
5.1 激活控制	(51)
5.2 编写事件过程	(52)
5.3 使用代码页	(53)
5.4 过程事件分类	(54)
5.5 使用事件过程	(57)
5.6 实例设计	(60)
5.7 生成可执行文件	(62)
5.8 获取硬拷贝	(63)
第六章 Visual Basic 语言	(65)
6.1 基本概念	(65)
6.2 常量和变量	(68)

6.3	用户自定义类型	(73)
6.4	逻辑流	(74)
6.5	过程	(79)
第七章	函数	(84)
7.1	字符串函数	(84)
7.2	转换函数	(87)
7.3	数学函数	(91)
7.4	日期函数	(94)
第八章	命令钮和菜单	(98)
8.1	命令钮及其响应	(98)
8.2	建立菜单驱动应用	(102)
8.3	增加和删除菜单命令	(109)
8.4	加速键与访问键	(115)
第九章	信息输入	(117)
9.1	文本框	(117)
9.2	实例设计	(118)
9.3	用任选钮进行选择	(122)
9.4	用检查框进行选择	(125)
9.5	用列表框进行选择	(127)
9.6	使用滚动条	(129)
9.7	用组合框进行选择	(130)
9.8	最简单的输入方法 InputBox \$	(133)
第十章	信息输出	(134)
10.1	简单的 MsgBox 输出	(134)
10.2	使用标签框输出	(136)
10.3	用文本美化窗体和图像框	(137)
10.4	使用多字体特性	(138)
10.5	表格输出	(140)
10.6	自动格式化数值、日期和时间	(141)
10.7	打印输出	(149)
10.8	间接打印机输出	(149)
第十一章	图形设计基础	(150)
11.1	Visual Basic 坐标系统	(150)
11.2	画图画	(154)
11.3	图像属性	(157)
11.4	画图图元	(160)
第十二章	数值应用	(164)
12.1	基数变换计算器	(164)
12.2	计算平均和标准方差	(167)

12.3	建立一个正弦和余弦值表·····	(169)
12.4	使用 Shell 分类排序·····	(174)
第十三章	图像和图形应用设计·····	(178)
13.1	安装一幅图像·····	(178)
13.2	从项目加载图像·····	(180)
13.3	动画例子·····	(183)
13.4	直方图·····	(185)
13.5	访问 Windows GDI 图形图元·····	(189)
第十四章	系统资源·····	(193)
14.1	获取系统信息·····	(193)
14.2	读键状态·····	(196)
14.3	查看时间和日期·····	(198)
14.4	显示设备信息·····	(199)
14.5	用鼠标器作图·····	(202)
14.6	报警信息显示·····	(205)
第十五章	金融应用·····	(210)
15.1	零存整取计算·····	(210)
15.2	取款计算·····	(212)
15.3	计算折旧·····	(214)
15.4	分期付款计算·····	(219)
15.5	抵押分期偿还表·····	(223)
第十六章	数据库应用·····	(230)
16.1	记录簿·····	(230)
16.2	人事数据管理·····	(233)
16.3	旅游机构预订系统·····	(237)
第十七章	绘制数学方程·····	(245)
17.1	矩形坐标系统·····	(245)
17.2	极坐标系统·····	(248)
17.3	绘制阻尼波形·····	(254)
17.4	使用付里叶级数·····	(256)
17.5	曲线拟合·····	(263)
第十八章	表示线、直方图和饼图·····	(272)
18.1	表示数量线图·····	(272)
18.2	表示数量直方图·····	(277)
18.3	表示数量饼图·····	(285)

第一章 Visual Basic 概述

1.1 什么是 Visual Basic

Microsoft Windows 是目前微机上最强有力、最受欢迎的操作系统,在 Windows 环境下提供了非常丰富的功能,例如程序管理器、文件管理器和打印管理等。但是在 Windows 下开发应用是十分困难的,Visual Basic 就是为了弥补这一缺憾而设计的,它是迄今为止最受青睐的 Windows 应用开发环境,它包含一系列的图形工具和高级语言指令,可以十分快速和方便容易地进行程序设计。

使用 Visual Basic 使我们的程序设计集中在“做什么”而不是“怎么做”,甚至没有任何经验的初学者都会得心应手,设计 Visual Basic 应用的三个基本步骤如下,在前面两步中不需要写出代码:

- 1) 画出组成用户界面的对象(Object)。
- 2) 设置每个对象的属性以改变它的行为。
- 3) 给每个对象附加程序代码。

在设计各个阶段中,Visual Basic 都提供了反馈和在线调试工具。

使用 Visual Basic 需要两点基本要求,第一,要求对 Microsoft Windows 有一般的了解,例如操作鼠标器、下拉式菜单和任选项等。第二点是具有一般的高级语言(如 BASIC、C 或 PASCAL)的程序设计经验,如熟悉 If... Then... Else 条件选择语句,For 循环语句,常量、变量和子程序等基本概念。但是,如果我们已经熟悉了 Microsoft QuickBASIC,在使用 Visual Basic 之后,就会发现 Visual Basic 是多么方便。

所有的 Windows 应用都有三个共同特征,它们都有一个称为 GUI 的公共图形用户界面;共享同样的硬件,如显示器和打印机等;共享 CPU。由于这些共同特征,使得单个程序的执行和终止不再是灵活的,因为之后要执行下一个应用,如此等等。这种动态的 Windows 环境要求每个应用以事件驱动地方式执行。

这样就出现了事件驱动程序设计,并取代了编写从顶部到底部执行的程序设计方法,我们设计的应用是来响应事件的,这些事件如按鼠标器按钮等都是由用户产生的,Windows 也产生一些事件,如当两个应用同时访问打印机时,Windows 便确定哪一个应用等待,当前一个应用打印完毕后,Windows 告诉等待的应用可以开始打印了。

当一个空的应用正在等待一个事件时,它仍然保留在这个环境中,我们可以运行其它应用,执行数据输入,改变窗口尺寸或者定制系统设计等。空应用的代码总是在等待,它准备在我们返回到该程序时激活。

Visual Basic 设计的应用也遵循上述应用的特征,它为应用开发提供了完美的设计工具,以便高效地生成、调试和测试应用,Visual Basic 支持的特征如下:

- 1) 一个完整的用于指向、按钮和拖动对象生成用户界面的工具箱。
- 2) 预先定义的命令对象允许我们建立命令和单选按钮、标签框、检查框、文本框、列表和组合框、计时器、菜单、垂直和水平卷滚条、图像框、窗口框架以及驱动器、目录与

文件选择框。

- 3) 一个属性条,它使得十分容易编辑每个对象的内部属性而不要写代码。
- 4) 在任何对象上快速双击(double-click), Visual Basic 自动显示相关的代码页,它们处理所选择项的所有事件。
- 5) 唯一的图形语句。
- 6) 一个调色板,用于定义我们的用户界面的颜色。
- 7) 一个完整的数学库。
- 8) 整数和长型整数数据类型。
- 9) 浮点数据类型和函数。
- 10) 一个货币数据类型,用于精确的商业计算。
- 11) 变量和定长度字符串数据类型。
- 12) 任何数据类型的可达 60 维的静态的动态数组,包含用户定义类型的数组。
- 13) 支持顺序和随机访问文件。
- 14) 一个允许我们建立水平菜单条的菜单设计窗口,这些菜单条中可以包含加速键,键盘访问键和灰色的或检查的菜单,所有这些菜单均不必写任何代码。
- 15) 一个包含大约 400 个图标(icon)的图标库,我们可以利用 IconWorks 样本程序使用或修改这些图标把它们加到每个应用中。
- 16) 直接访问所有的 Windows 函数,包括经由 Windows 的动态数据交换(DDE)来进行内部应用通信。
- 17) 在线调试和解释每个所编写的语句,直接把代码转换成可执行的窗体(Form)。
- 18) 调用 Windows API 函数。
- 19) 能够把每个应用转换成紧缩 EXE 文件,它可以脱离 Visual Basic 直接运行。
- 20) 面向对象的程序设计方法。

1.2 安装和运行 Visual Basic

Visual Basic 系统盘有 2 张 5.25 寸(1.2MB)磁盘和 3 张 3.5 寸(720K)磁盘,其中的 SETUP.EXE 是一个安装程序,它也是用 Visual Basic 编写的,该安装程序不仅用于安装 Visual Basic,还可以有选择性地加载 Help 系统,学习与样本应用以及图标库。

Visual Basic 的运行环境为 80286 或更高微处理器,并有 1MB 内存和 Windows 3.0 或更高版本,还需要一个硬盘,一个 5.25 寸或 3.5 寸的驱动器和一个鼠标器。

一旦这些软件环境和硬件环境齐备后,我们便可以开始安装 Visual Basic 了,在 Windows 的文件管理器中键入 a:setup 或者在 DOS 下输入 a:win setup 来运行 SETUP,然后选择在硬盘上安装 Visual Basic 的子目录,再建立这些子目录并复制文件。

在 Visual Basic 系统安装好后,我们可以采用 3 种方法之一启动它。如果我们从 Windows 启动,只需在 Visual Basic 图标上按鼠标器按钮;如果在文件管理器中,可以从文件管理器选择 Run 命令并键入 VB 即可。为了从 DOS 下启动 Visual Basic,输入 win vb。

我们建议在启动 Visual Basic 时把 Program Manager 设置成最小,这样避免在同时打开太多窗口时覆盖混乱,为此,在 Program Manager 中打开 Options 菜单并选择 Minimize on

USE 可以把程序管理器设置成最小。

在使用 Visual Basic 时可以进行在线帮助,这是通过调用 Help 实用程序实现的,它查找 Help 数据库回答用户的任何提问。Help 的基本操作如下:

Help 实用程序是一个上下文有关的系统,它知道我们所处的窗口和代码行,或者我们使用的哪个 Visual Basic 任选项。按 F1 功能键后立即在 Help 窗口中显示任何有效命令或任选项描述,如图 1.1 所示。

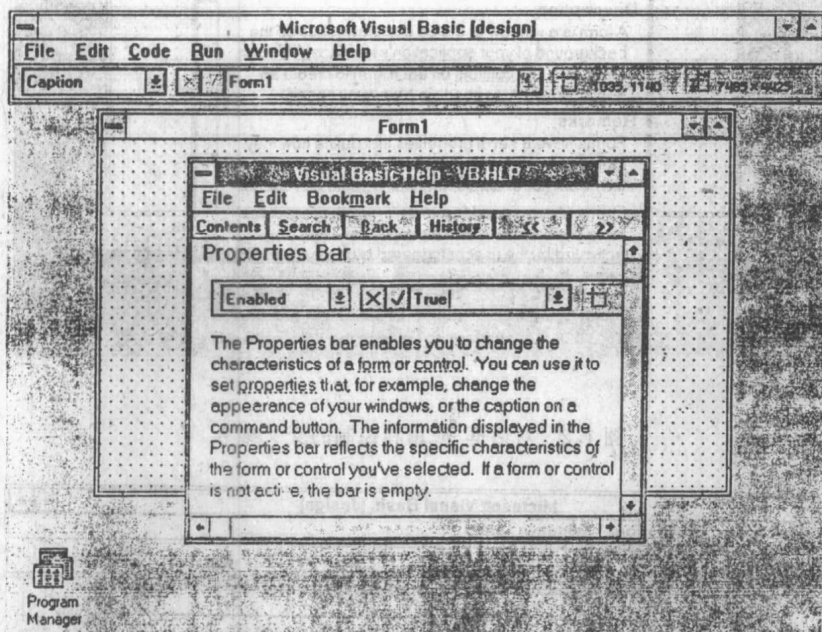


图 1.1 在属性条上进行在线帮助

我们看到图 1.1 中的窗体、控制(Control)和属性(Properties)项,在 Help 窗口中任何高亮度或下划线显示的项都可以作为一个标题进行交叉参考,如果一个项的下划线为破折号,则交叉参考是一个简单的定义,带有实线下划线的项产生完整的交叉参考文本,我们可以选中高亮度的项或者按鼠标器按钮打开子标题窗口(如图 1.2 所示),还可以使用 Forward 和 Backward 按钮查阅我们的查询标题(topic-query)的历史列表。

在 Help 窗口中的垂直和水平滚动条用于向上或向下扫描解释特殊特征的例子文本,在很多情况下,Help 窗口显示一个代码例子,如图 1.3 所示,我们可以复制并运行这个例子程序。

除了激活 Help 外,我们还可以使用标题索引(Index of topics)或 Search 任选项找出与问题有关的内容。在 Search 任选项上单击鼠标器按钮打开一个对话框,在该对话框中输入一个搜索内容,例如 Form,如图 1.4 所示。在我们输入一个搜索标题时,Search 实用程序试图用我们输入的每个新字母与所有相关标题进行匹配,在输入完毕后,Select a topic 下方的 Search 对话框的底部显示相关归类的列表。

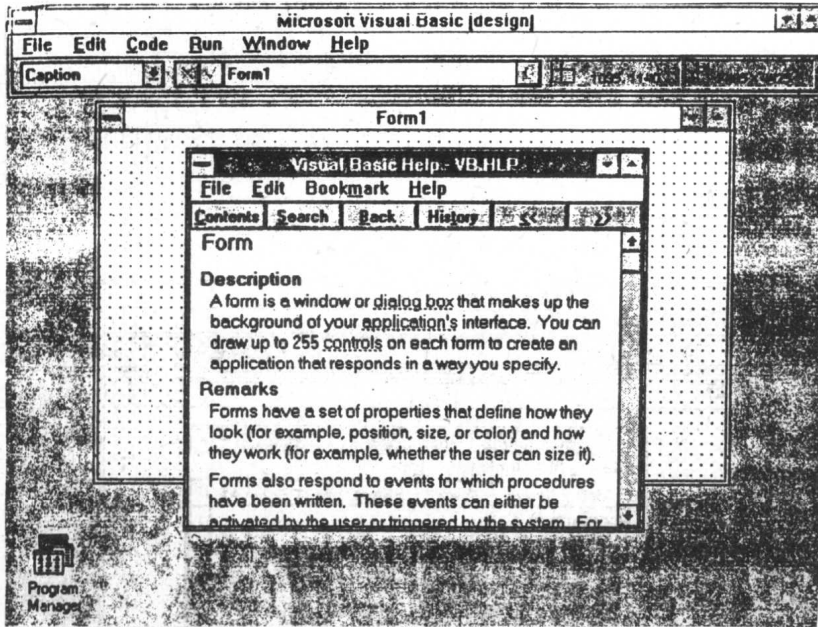


图 1.2 对话框、应用和控制的交叉参考

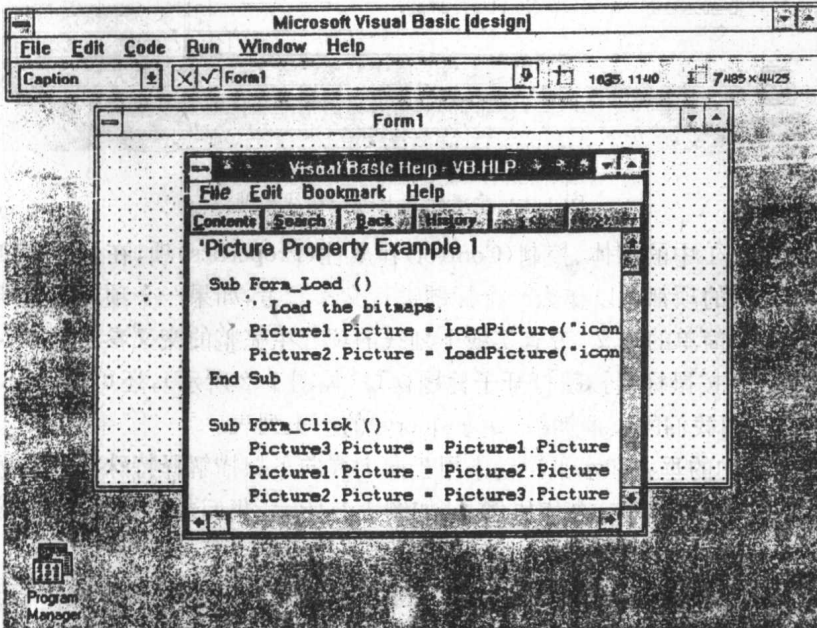


图 1.3 Picture 属性帮助例子程序

为了从 Help 窗口运行一个代码例子，必须首先打开一个窗体，它可以是一个现存的窗体或者是一个特定打开的测试该例子代码的窗体。打开 Help 窗口，作一次 Circle 方法的搜索，移到 Help 窗口，在 Circle example 关键字上单击鼠标器按钮，如图 1.5 所示。当显示代码后，在 Edit 菜单的 Copy 任选项上单击，再在该窗体上双击，则显示该窗体的代码窗口，选择

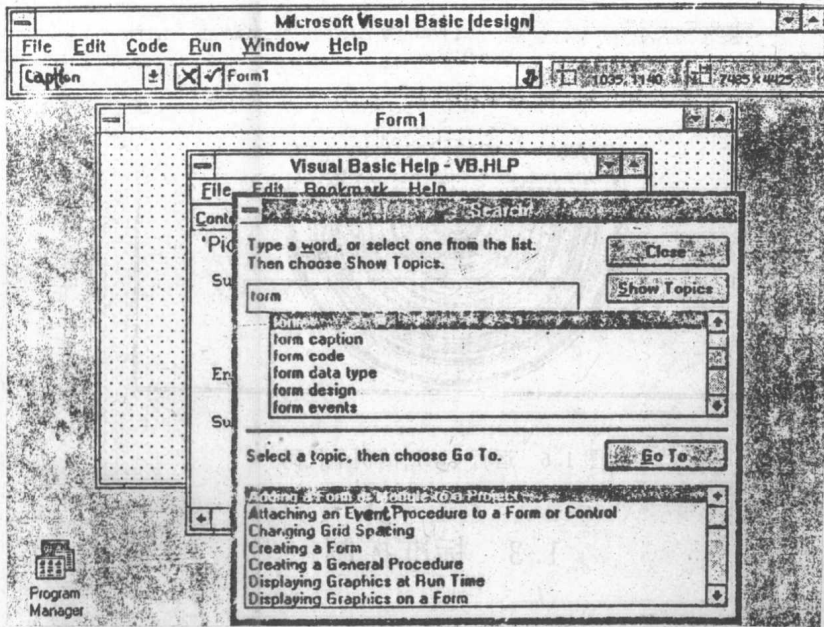


图 1.4 搜索 Form 标题

Edit|Paste, 最后一步是在该窗体上双击并插入一个子过程 CircleDemo 的调用。

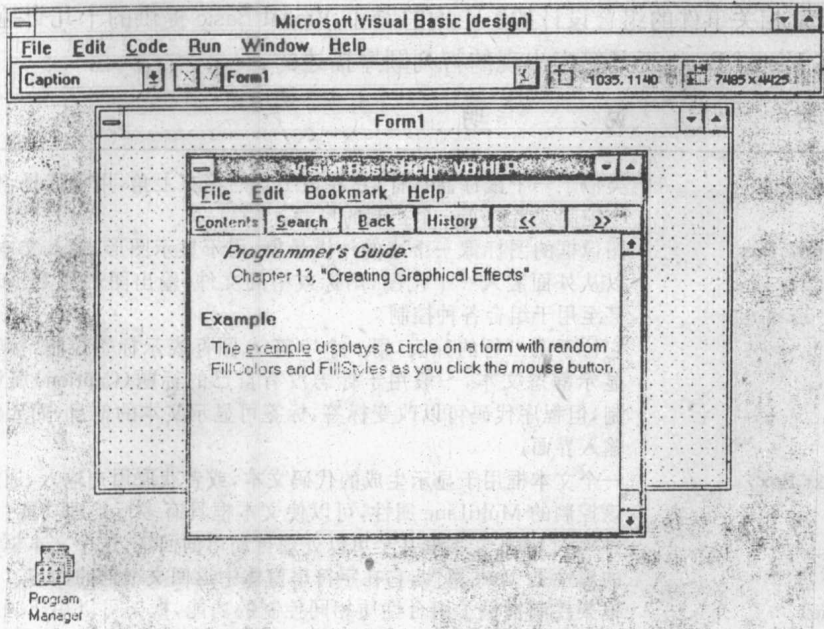


图 1.5 获取 Circle example 代码

从 Run 菜单选择 Start 任选项或者按 F5 功能键运行这个程序, 在窗体上单击测试实际的代码, 其运行结果如图 1.6 所示。当我们操作完毕, 选择 Run|End 命令。

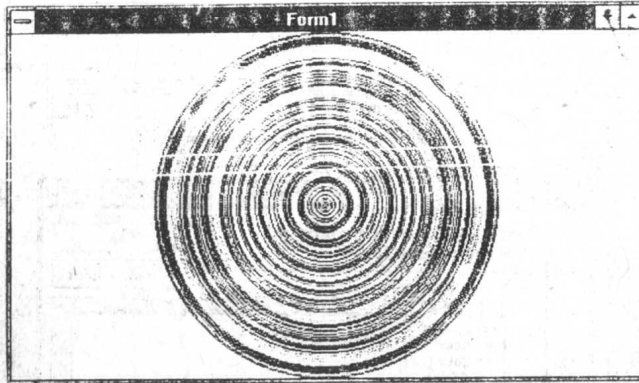


图 1.6 运行 Circle 的例子代码

1.3 标准控制

Visual Basic 中的窗体(Form)用于存贮一个应用看得见的元素及其相关代码。窗体窗口提供设计用户界面的窗口或对话框,每一个窗体窗口最多可容纳 255 个控制。所谓控制(control)就是放在窗体上的图形对象,它的主要功能是用来接受输出数据,作为用户与计算机间的界面。每个控制都有自己的可认识的属性和事件集,所有的 Visual Basic 界面都是使用这些控制及其相关事件的组合设计的。下面总结了 Visual Basic 提供的十几个控制,每个控制以它的在 Visual Basic 工具箱中出现的行列顺序描述的:

控制名	说 明
指针(Pointer)	类似于一个鼠标器指针,它在该工具的箱左上角,用于选择,移动一个控制或者改变一个控制的尺寸。
图像框(Picture Box)	图像框的图标像一个小沙漠景象,用于显示图形,输入文本,还可以从外面装入一个位图,图标或中间文件,输出图形和 Print 方法,甚至用于组合各种控制。
标签(Label)	它位于第二行的开头,用一个字符 A 图标表示标签控制,标签用于显示静态文本,一般用于标志没有自己的标题(Caption)属性的控制,但程序代码可以改变标签,标签可显示文本的信息,而无法作为输入界面。
文本框(Text Box)	一个文本框用于显示生成的代码文本,或者获取用户输入,通过改变该控制的 MultiLine 属性,可以使文本框具有多行处理功能,当在这种模式下时,文本框甚至可以处理自动单词回绕。这个文本框最重要的属性是 Text 属性,它在字符串窗体中返回文书框的内容。
框架(Frame)	框架控制提供了组合性质相同控制的功能,框架可以用于画简单的框。当把其它控制放在一个框架中时,在一个窗体设计阶段,如果移动该框架,那么包含在该框架中的所有控制也跟着移动。
命令钮(Command Button)	命令钮像键盘上的一个键,在用户激活它时执行一个特定动作,有时也称为下推钮(push buttons),当用户选择时其图形便凹陷,命令钮控制通常由相关的 Command_Click()事件激活。它有两个唯一的属性 Default 和 Cancel,它们产生经常看到的 OK 和 Cancel 钮,分别响应 Enter 和 Esc 键。

控制名	说 明
检查框(Check Box)	检查框供用户选择 True/False 或者 yes、no 控制。当用户选择了对应的任选项时,该检查框显示一个大的 X。如果再击即取消选择则又变成空白的检查框。检查框不同于任选项控制,它可以从一组任选项中选择多个检查框。
选择钮(Option Button)	有时也称为单选钮(radio button),它用于从一组任选项中选择一项,不同于检查框可以作多种选择,多个相关控制中这里只能选择一个项。
组合框(Combo Box)	组合框是列表框和文本框的组合,它可由输入方式或选择方式等输入数据。一个组合框的 Style 属性允许我们选择如下 3 种形式之一:下拉式组合框(Style=0);简单的列表显示(Style=1);下拉式列表框(Style=2)。
列表框(List Box)	列表框控制显示一个项的列表,用户可以从中选择某个项,List 属性定义为一个 String 类型的数组,我们可以使用 ListIndex 属性访问 List 数组,ListCount 属性返回数组中的行数。组合框控制和所有的文件系统控制使用相似的属性。
滚动条(Scrollbars)	滚动条有垂直滚动条(Vertical Scrollbars)和水平滚动条(Horizontal Scrollbars),前者可在列表区域窗体中显示数据的位置,后者可在一列表区域或窗体中显示数据的位置。
计时器(Timer)	计时器控制用于在时间间隔内激活特定的事件,使用 InterVal 属性(单位为毫秒),计时器控制可以用于产生报警,在背景中运行过程或者显示其它时间相关事件的坐标。
驱动器列表(Drive List)	当一个应用界面使用了一个驱动器列表控制时,它向用户显示当前所有可用的驱动器,用户可以从中选择一个,它可以与目录列表框和文件列表框一起使用用于指定完整的路径。
目录列表框(Directory List)	使用目录列表框控制,用户可以浏览磁盘的目录层次,它用于寻找或切换当前磁盘驱动器上的目录及路径。
文件列表框(File List Box)	文件列表框提供了查看指定目录中所有文件和执行高级通配符搜索的功能。应用能够使用 FileName 属性确定用户选择哪些文件,通常与目录列表框控制一起使用,文件列表框保证以最少的代码写出一个文件打开过程。

上面列出了 Visual Basic 提供的标准控制,在一个窗体上有两种建立控制的方法,以下分别讨论这两种建立控制的方法。

第一种是将鼠标器指针在工具箱上适当的标准控制上单击一下,鼠标器指针在窗体上变成了一个“+”号,而且工具箱上的该控制变成黑色,此时将鼠标器指针移动到窗体中适当位置拖动出一个大小合适的控制。所谓拖动(drag),即按住左边按钮不放移动鼠标器。例如我们把鼠标器指针移到工具箱的文本框上单击一下,然后拖动该文本框产生出如图 1.7 所示的文本框,再放开按着的鼠标器按钮,使形成了一个适当的文本框控制。这时在该控制的四周出现 8 个黑色小方形,表示此控制是活动控制(Active Control),控制周围的 8 个小方块。除了表示文书控制是活动控制外,还表示可以改变其尺寸,采用的方法是用鼠标器拖动其上的上下左右 4 个黑色小方块使该控制朝指定方向变大或缩小。在完成控制之后,如果要改变该控制的位置,只需将鼠标器指针放入到这个控制之中,然后就可以拖动该控制到外移动,这样就可以改变控制的尺寸和改变控制的位置。

第二种建立控制的方法是在所选定的工具箱上双击选中的控制,则在窗体正中央处出现一个控制,如图 1.8 所示。

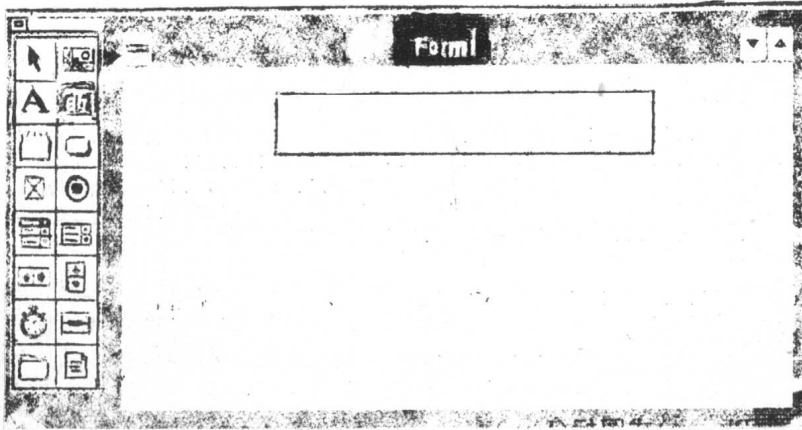


图 1.7 建立文本框控制

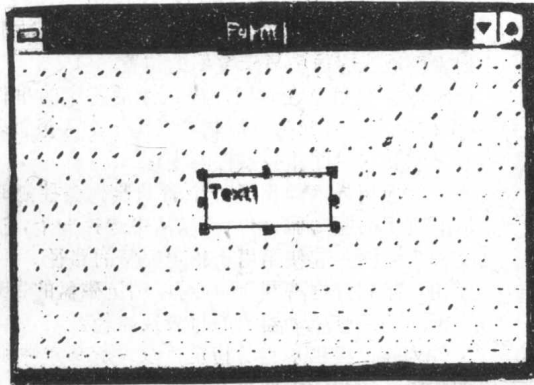


图 1.8 再现在正中央的文件框

如果要复制较多的控制,便可以采用“拷贝”方法,其操作如下:首先采用前面的方法把要复制的控制变成活动控制,其周围出现黑色小方块,如果不是活动控制,就要单击该控制。然后单击菜单条 Edit 之上的 COPY 任选项,这时 Visual Basic 将该控制拷贝到 Windows 的剪接板(Clipboard)上。然后单击命令 Paste,这时出现一个对话框显示“You already have a control named “Text1”, Do you want to create a control array?”,其意义是说已经有一个名称为“Text1”的控制,是否要建立一个控制数组。因为这里不需要建立控制数组,故回答“NO”,这样在该窗体的左上角出现一个与活动控制完全相同的控制。

清除控制的操作是先将该控制变成活动控制,再按 DEL 功能键,这个控制便立即从窗体中消失了。

我们还可以利用属性条上的有关任选项改变控制或窗体的尺寸和位置,这些任选项有“Width”、“Height”、“Top”和“Left”等。

1.4 控制属性

一旦我们使用前面的各种控制对象设计了窗体界面,便可以指定它们的行为,如图 1.9

所示显示了用于访问控制功能的属性条(Properties bar)。

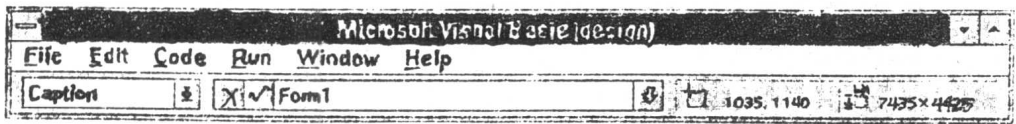


图 1.9 属性条

我们在 Visual Basic 中建立的每个对象都有一个与之相关的称之为属性的特征集合,最常用的属性集合定义一个对象的尺寸,屏幕位置和颜色。每次建立一个对象,Visual Basic 便赋给它适当的属性并把它们初始化为预先定义的值,属性条包含适合于选择的对象的属性列表,并允许我们的改变它们的值。例如,如图 1.10 和 1.11 所示显示了改变命令控制字体尺寸后的结果。

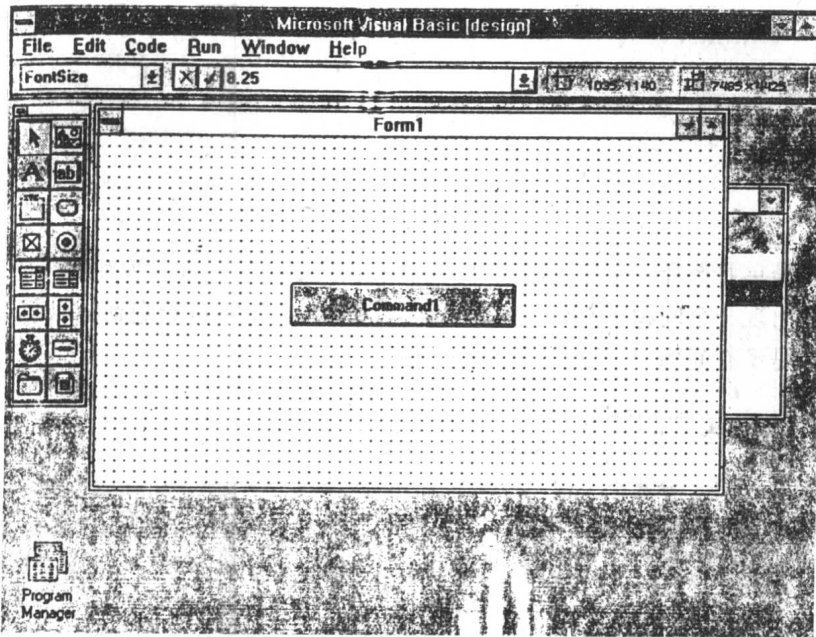


图 1.10 使用 8.25 点缺省字体尺寸的 Command1

每次我们把一个控制加到一个窗体中时,Visual Basic 自动给出名称和连续编号。例如,我们使用两个文本框,Visual Basic 自动为它们分别取名为 Text1 和 Text2,命令钮取名为 Command1,框架取名为 Frame1,等等。

1.4.1 改变控制属性

我们通过一个例子讨论改变控制属性的方法。把鼠标器移到工具箱的第 3 行第 2 列,然后单击命令钮图标,现在通过在窗体中定位鼠标器,按住鼠标器左按钮不放,并改变其尺寸来在窗体中建立一个命令钮对象。

现在进入属性条,其操作是单击 Caption 属性右边的列表命令箭头,翻动属性列表直到出现 FontSize,并单击它,属性条的中央部分现在显示所选择的属性为当前设置即 8.25 点。

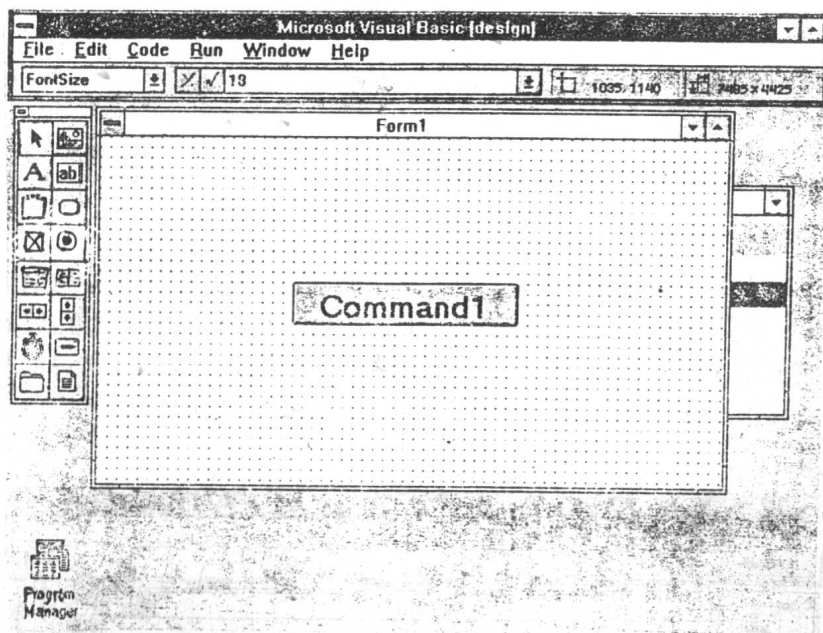


图 1.11 使用 18 点字体尺寸的 Command2

单击设置列表简单并选择一个更大的尺寸,注意现在控制的文本立即发生改变。

这样可能出现一个问题?如果我们选择的尺寸变大,控制不可能宽到足以包含所有更大的文本。答案是没有问题,只需简单地单击该控制并改变其尺寸。

1.4.2 改变属性的捷径

如果我们的应用十分复杂,在一个窗体中可能包含很多控制,如果采用前面讨论的方法改变每个控制的属性,这是十分花费时间的,属性条有一个有趣的特征可以帮助我们解决上述问题。一旦我们选择了一个对象属性,例如 FontSize,那么该属性一直保留让我们从一个对象到另一个对象。

如果我们要求窗体中的控制具有相同属性,选择其中的一个控制和特定属性,改变其设置,然后顺序选择其它控制,那么在属性条中对于每个对象所选择的属性是高亮度的,等待我们输入新的设置,下一个控制具有当前选择的属性。

1.4.3 键盘属性选择

属性条另一个方便的特征是能够快速跳到特定的属性。第一种方法是前面描述的,单击属性列表的向下箭头,上下扫描该列表,作出自己的选择。一种更快的方法是当属性出现后按要查找属性的第一个字母,例如,为了找出一个对象的 FontSize 属性,我们单击缺省的属性,并按 F4 键。每次我们重复一个字母,属性条使用同样的起始字母显示下一个命令,这里 FontSize 是以 F 开头的第 4 个属性。