

[美] John Socha & Devra Hall 著

夏春和 张玉亭等 译

岳晋生 校



Visual Basic 3.0 程序设计自学教程

- 无需程序设计经验便可轻松学习卓越的新一代程序设计语言 Visual Basic 3.0
- 学习如何创建全功能的 Windows 应用程序
- 学习如何将最符合特定需求的控制和最完备的功能引入应用程序中
- 随附软盘无偿奉送价值 75 美元的商用程序设计工具软件



水利电力出版社



Visual Basic 3.0 程序设计自学教程

[美] John Socha 著
Devra Hall

夏春和 张玉亭等 译
岳晋生 校

水利电力出版社

1995

内 容 提 要

本书是一本极有价值的参考用书。它通过实例循序渐进地讲解了 Visual Basic 3.0 的基本概念及程序设计技术。全书共分三个部分：第一部分对 Visual Basic 进行了全面介绍，其中包括：如何启动与退出 Visual Basic，变量与值，建立程序等。第二部分讲解如何建立较大的 Visual Basic 程序，其内容包括：设计与建立程序，读写地址簿，处理多个记录，存取数据，以及如何查找、打印、排序和删除操作。第三部分介绍了一些先进技术，并给出一定提示等等。此外，还有一代码库，供读者编写程序时使用。

本书适用于计算机编程人员、技术人员及广大的计算机用户，同时也适用于各大专院校师生及计算机爱好者。

本书英文版由美国 MIS:PRESS 公司出版。本书的中文版于 1994 年 9 月经美国远东图书公司(Far East Books, USA)授权水利电力出版社在华独家出版。未经出版者许可，任何人不得以任何形式、手段复制或抄袭本书的内容。

本书的中文翻译、审校及文字处理工作，由美国远东图书公司完成。

Copyright © 1994 by MIS: PRESS

书 名	Visual Basic 3.0 程序设计自学教程
作 者	John Socha [美] Devra Hall 著
译 校 者	夏春和 张玉亭等 译 岳晋生 校
出版、发行	水利电力出版社(北京三里河路 6 号) 各地新华书店经售
排 版	五环出版服务部
印 刷	北京市朝阳区小红门印刷厂
规 格	787×1092 mm 16 开本 21.25 印张 509 千字
版 次	1995 年 4 月第一版 1995 年 4 月北京第一次印刷
印 数	0001—5000 册
定 价	69.00 元(含磁盘)
书 号	ISBN 7-120-02172-9/TP·82

简介

如果要建立自己的程序,但不知如何去做;如果要编写 Windows 程序,但不知如何用 Windows 编程,怎么办?现在,Visual Basic 已改变了这一状况。

首先,编写 Windows 程序并建立 Windows 屏幕实际上是一件非常令人愉快而简单的事情。读完这本书之后,读者就能够在若干小时内用 Visual Basic 建立有经验的专业程序员用任意其他计算机语言要耗费几天(甚至几个星期)才能编写出的程序。即便读者只是有兴趣为自己编写程序,Visual Basic 也定会令人满意,而读者无需计算机专业的文凭或任何编程背景。

在本书中,我们将教会读者如何用 Visual Basic 编写程序。其他有关 Visual Basic 的书籍也会这样做,不过,我们将涉及一些在其他书籍中无法找到的内容:我们将用地址簿程序作为示例程序教会读者如何使用 Basic 语言,以及如何设计并建立实用程序。本书的作者已有多年软件开发的经验:John 设计并编写了由 Symantec 购买且十分畅销的 Norton Commander (Norton Desktop for DOS 的父代),而 Devra 负责开发过 hypermedia 软件工具及应用程序,如 IBM 的 Illuminated Books and Manuscripts。

在这一简介中,我们将大致介绍一下本书,并介绍一下本书的读者对象。

0.1 能用 Visual Basic 编写真正的程序吗?

读者也许担心,是否能用 Visual Basic 编写真正的程序?这毫无问题。我们已见过用 Visual Basic 编写的许多实用程序,且许多公司正致力于用 Visual Basic 编写家用应用程序。这是为什么呢?这是因为用 Visual Basic 编写程序所用时间少且无需很多经验。再者,只要读者努力钻研本书,便可用 Visual Basic 完成用任意其他语言可以完成的工作。

0.2 本书适用于哪些人?

不管读者是初学者还是经验丰富的程序员,都会发现本书中的某些章节适用于他们。

0.2.1 对初学者

本书的前两部分完全针对那些以前从未编写过程序,或不知晓何为 Basic 语言的读者。我们使用一种非常文雅且直观的交互式方法讲解 Basic 语言,从第一章开始,读者便会接触到非常简单的单行 Basic 程序。通过运行并试验程序,读者便会掌握每个新概念。我们不使用填鸭方式,而使读者能够一次掌握若干新概念。

在第二部分中,我们将建立一个地址簿程序,使读者易于随时更改地址簿。该程序是一个非常有用的学习工具,且本身适用于作为程序使用。

一旦读者读完本书的前两部分,就可以编写自己的程序了。读者会发现,本书的其余部

分极有价值且具参考性,而且,读者可以回过头来,阅读各个章节,以解答特殊问题。

0.2.2 对新接触 Basic 的程序员

如果读者已经是程序员,但新接触到 Basic 或 Visual Basic,也许要浏览一下前几章。读者也会发现,本书后面章节中的代码和想法具有很大价值。

0.2.3 对新接触 Visual Basic 的 Basic 程序员

如果读者已经是 Basic 程序员,但从未使用过 Visual Basic,那么可以浏览一下前几章。读者也许会发现,用 Visual Basic 编写程序与用其他版本的 Basic(如 Quick Basic)编写程序有很大不同,这是因为 Visual Basic 具有很多的视屏功能。画一个界面,之后编写代码,使界面极为生动。另一方面,许多 Quick Basic 程序员习惯于在项目以后而不是开始便建立界面。我们认为,一旦读者掌握了视屏方法,就会发现这种方法是极为有用的。读者还会发现,本书的其余部分作为参考以及处理 Windows 特定任务的源代码和算法有很大价值。

0.2.4 对高级程序员

本书第三部分和随附的磁盘适用于所有程序员。想学习如何编程的任何人,如果拥有为其需要可使用和修改的一定量元素,就会使其作业更为容易。第三部分拥有一定量的提示、技术及进一步信息。此外,还有一个完整的代码库,供读者在自己的程序中使用。所有这些代码均包含在本书后面的磁盘中。

本磁盘还包含一定量的全功能定制控制和函数,读者可在自己的程序中使用。它们有可获利的质量,由三个公司提供并作为 Visual Basic 附加产品。这三个公司是:Crescent Software 公司、MicroHelp 公司和 Sheridan Software Systems 公司。

0.3 磁盘中的内容

本书后面有一张磁盘,该磁盘含有可在读者自己的程序中使用的代码和工具。下面是磁盘中的内容:

- 示例。读者会发现本书示例的代码。
- 代码库。从本书第三部分开始,读者会发现代码库,因此,读者不必键入任何代码。
- 定制控制。拥有一定量的定制控制,可在读者自己的程序中使用。这些定制控制来自商用软件库(由 Crescent, MicroHelp 和 Sheridan 提供)的全功能示例,且其本身使本书具有一定价值。在读者所建立的任何程序中可随意使用这一软件。
- 全文档。每个公司都为其软件提供了文档,但该文档太长,无法添加到本书中。因此,我们将这些文件放入磁盘中,作为可以打印的 Windows Write 文件。

目 录

简介

第一部分 Visual Basic 介绍

第一章 Basic 初步	(3)
1.1 启动 Visual Basic	(3)
1.2 退出 Visual Basic	(3)
1.3 什么是程序设计	(5)
1.4 使用 Debug 窗口	(5)
1.5 离开运行方式	(7)
1.6 Print 命令	(8)
1.7 Basic 命令的构成	(8)
1.8 执行算术操作	(11)
1.9 相关工具	(15)
1.10 小结	(15)
第二章 变量与值	(17)
2.1 什么是变量?	(17)
2.2 变量与类型	(20)
2.3 相关工具	(26)
2.4 小结	(27)
第三章 在屏幕上画图	(28)
3.1 在窗口中画图	(28)
3.2 画图与缇(Twips)	(30)
3.3 对象与窗体	(31)
3.4 再谈画图	(32)
3.5 相关工具	(35)
3.6 小结	(36)
第四章 建立程序	(37)
4.1 建立第一个程序	(37)
4.2 建立 Sketch 程序	(41)
4.3 If..Then 命令	(47)
4.4 完成 Sketch	(51)
4.5 相关工具	(52)
4.6 小结	(53)
第五章 添加菜单条	(54)

5.1	建立菜单条	(54)
5.2	建立菜单标题	(55)
5.3	完成菜单条	(58)
5.4	上托菜单	(63)
5.5	小结	(64)
第六章	用数组变量保存线	(65)
6.1	设计新的 Sketch	(65)
6.2	数组变量	(66)
6.3	保存 Sketch 中的点	(69)
6.4	打印图像	(77)
6.5	建立 EXE 程序	(79)
6.6	最后的 Sketch 程序	(79)
6.7	相关工具	(82)
6.8	小结	(83)
第七章	建立时钟程序	(85)
7.1	设计图标时钟	(85)
7.2	处理图标	(86)
7.3	读时钟	(88)
7.4	使用计时器	(89)
7.5	建立计时器	(89)
7.6	设置计时器	(91)
7.7	显示时间	(92)
7.8	绘制表盘	(95)
7.9	最小化图标时钟	(98)
7.10	相关工具	(107)
7.11	小结	(108)

第二部分 建立较大型程序

第八章	设计并建立程序	(111)
8.1	如何设计程序	(111)
8.2	设计用户接口	(112)
8.3	建立程序	(116)
8.4	小结	(119)
第九章	建立地址簿的接口	(121)
9.1	建立控制	(121)
9.2	控制与属性概述	(129)
9.3	完善地址簿	(131)
9.4	设置组合框	(132)

9.5	相关工具	(136)
9.6	小结	(136)
第十章	读写 Address Books(地址簿)	(138)
10.1	处理磁盘文件.....	(138)
10.2	数据包:用户自定义类型	(146)
10.3	读取和写入复合变量.....	(150)
10.4	去掉字符串尾空.....	(152)
10.5	使用联机帮助.....	(153)
10.6	创建新的子例程.....	(154)
10.7	相关工具.....	(158)
10.8	小结.....	(159)
第十一章	对多个记录的处理.....	(160)
11.1	建立模块.....	(160)
11.2	多记录方式的 DATABASE.BAS	(166)
11.3	向窗体中添加多个记录.....	(171)
11.4	支持 Cut、Copy、Paste 和 Undo 命令	(182)
11.5	相关工具.....	(183)
11.6	小结.....	(183)
第十二章	检索、打印、排序和删除.....	(185)
12.1	检索.....	(185)
12.2	打印地址.....	(187)
12.3	地址排序.....	(188)
12.4	删除地址.....	(188)
12.5	小结.....	(188)
第十三章	数据存取:数据管理器和数据控制	(189)
13.1	建立地址簿数据库.....	(189)
13.2	创建带有数据敏感控制的窗体.....	(196)
13.3	创建菜单条.....	(203)
13.4	编写代码.....	(203)
13.5	小结.....	(216)

第三部分 先进技术工具箱

第十四章	先进技术简介.....	(219)
14.1	Visual Basic 特有技术	(219)
14.2	定制控制.....	(219)
14.3	DLL 与 Windows 函数	(220)
第十五章	不同类型的显示器对程序的影响.....	(223)
15.1	屏幕分辨率.....	(223)

15.2	逻辑英寸	(226)
15.3	如何得到轮廓清晰的图像	(227)
15.4	如何改变 Visual Basic 对象的大小	(228)
15.5	调整用户程序	(230)
15.6	日历钟例子	(236)
第十六章	控制工具箱	(246)
16.1	进展条	(246)
16.2	在控制周围画阴影	(248)
16.3	检查正文框输入	(249)
16.4	正文框中的覆盖方式	(253)
16.5	口令正文框	(254)
16.6	支持 Edit 菜单	(255)
16.7	设置 3-D 按钮的 Default 与 Cancel 属性	(258)
第十七章	窗体工具箱	(260)
17.1	窗体居中	(260)
17.2	使用内尺寸设计窗体大小	(261)
17.3	限制窗体的尺寸	(263)
17.4	重新设计窗体控制尺寸	(264)
17.5	显示启动屏幕	(267)
第十八章	画图工具箱	(269)
18.1	快速画线	(269)
18.2	快速尺画法	(272)
18.3	画填充多边形	(276)
18.4	调整打印机上的线条宽度	(279)
18.5	使用屏幕像素方式打印	(281)
18.6	以 twip 标度使用 API 调用	(283)
第十九章	字体工具箱	(287)
19.1	通过 TextHeight 申请字体	(287)
19.2	设置字体名称的可靠方法	(289)
第二十章	其它工具箱	(298)
20.1	多媒体声音	(298)
20.2	运行 DOS 程序	(299)
20.3	编写大程序	(302)
20.4	检查 DOS 和 Windows 的版本	(305)
20.5	查找程序所在目录	(307)
20.6	建立 Windows 3.1 屏幕保存程序	(308)
附录 A	安装 Visual Basic 及随书磁盘文件	(317)
附录 B	使用随书磁盘的商品化软件	(322)
附录 C	术语汇编	(326)

第一部分 Visual Basic 介绍

在这一部分中,读者将掌握某些基本编程概念,并向读者提供可以建立某些有用的 Basic 程序——即 Visual Basic 的基本概念。在第一、二章中,读者将了解什么是程序设计以及如何使用算术函数、变量、值和类型。在第三章,读者将掌握如何在屏幕上用 Visual Basic 画线。在第四章,读者将建立第一个 Visual Basic 程序 Sketch,并掌握自学 Visual Basic 的方法。在第五章,我们将向读者说明如何将菜单条添加到 Sketch 程序中。在第六章,读者将学会如何使用数组变量。最后,在第七章,我们会将所有 Basic(以及 Visual Basic)的知识合而为一,为读者自己的桌面建立一个时钟程序。

第一章 Basic 初步

- 介绍 Visual Basic
- 使用 Debug 窗口
- 掌握简单的 Basic 命令:Beep,End 和 Print
- 命令句法基础
- 计算机算术和优先级

在本章中,我们将大致介绍一下 Visual Basic。在我们进行了某些介绍并确保我们在同一个起跑线之后,我们直接切入正题,讲解如何编写程序。

我们所要介绍的程序设计的基本概念相对来说是较为简单的,在实际编写某些较短的 Basic 程序之前对此不必掌握许多。本章将介绍给读者的每个程序只有一行长,读者可以直接在 Visual Basic 中运行这些程序。因此,读者会看到这些程序的功能是什么。读者将会非常惊奇地发现,Basic 怎么会这样让人赏心悦目!

首先,我们需要概述若干预备知识。

1.1 启动 Visual Basic

读者要做的第一件事是确保已安装了 Visual Basic。如果尚未安装且 Microsoft 手册中的内容不够且需一定帮助的话,可在附录 A 中得到详细信息。在完成这项任务后,再回到此处。

此时,用户拥有一个称为 Microsoft Visual Basic 的程序组,它是由 Setup 程序建立的。当打开程序管理器(Program Manager)时,该程序组类似图 1.1 中的内容。

注意到左下角的小时钟图标了吗?这是一个显示当前时间的小程序。它类似 Windows 部分中的 CLOCK.EXE,但它看上去更亲切一些,且有秒针。不仅如此,它完全是用 Visual Basic 编写的。在本书后面,读者会看到完整程序!在本书随附的磁盘中还有一个增强版本。

双击 Microsoft Visual Basic 图标,将启动 Visual Basic。在以后的若干章节中,读者会经常见到图 1.2 所示的 Visual Basic 屏幕。

1.2 退出 Visual Basic

如果读者已使用过其他 Windows 程序,也许知道了如何退出 Visual Basic。但是,如果读者新接触 Windows,那么就要下拉 File 菜单(按 Alt+F)并选择 Exit 项(X)。

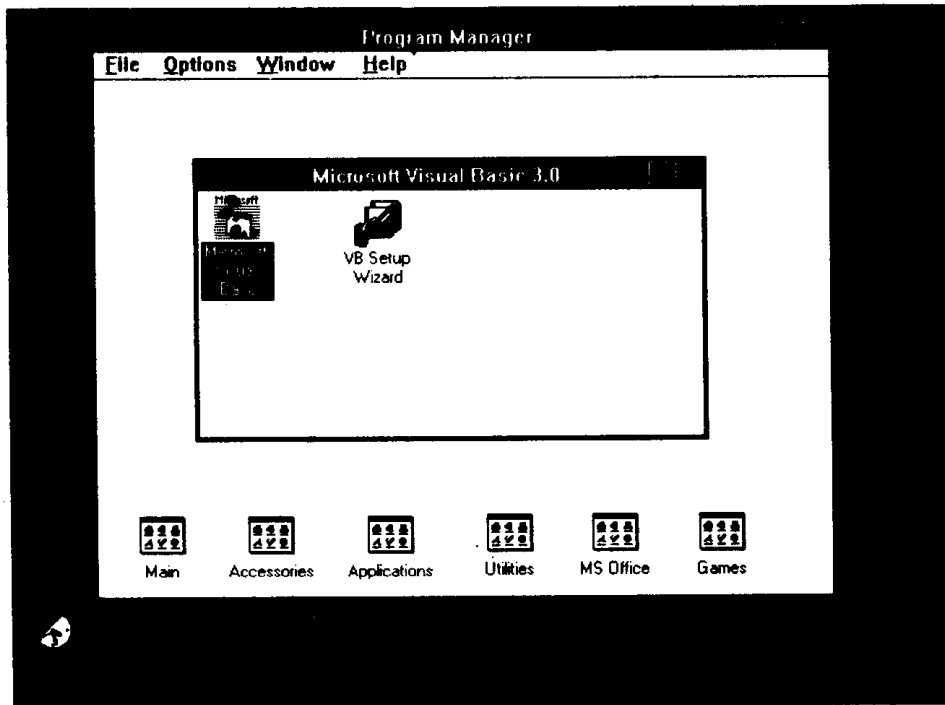


图 1.1 在安装 Visual Basic 后,用户会拥有一个含 Visual Basic 图标的程序组

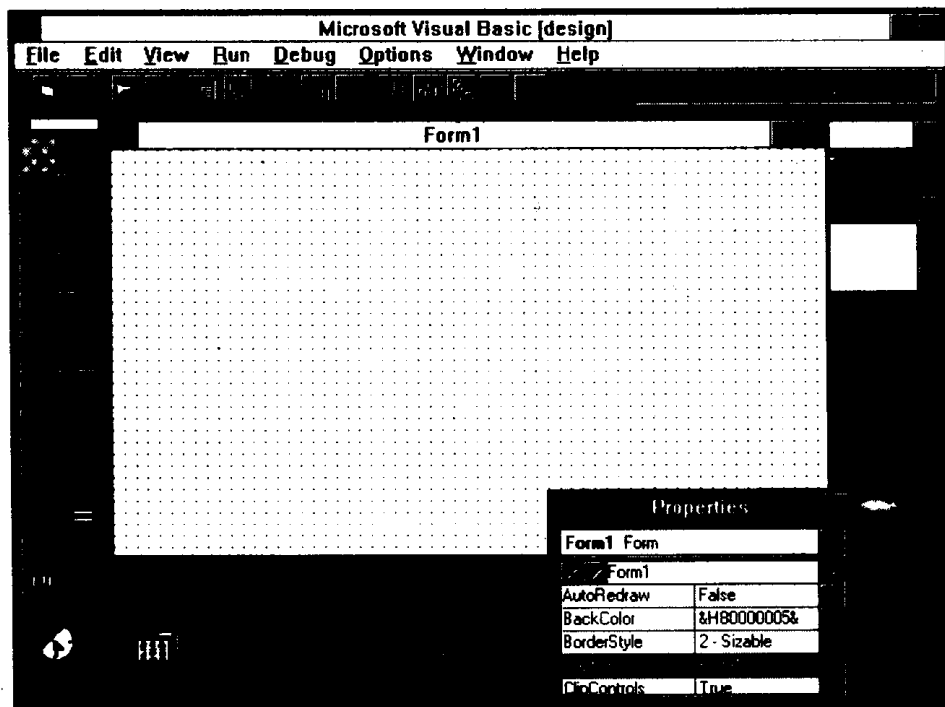


图 1.2 在首次启动 Visual Basic 时,会看到类似图中所示的屏幕(Options 菜单中的 Program Manager's Minimize on Use 选项用于启动 Visual Basic 后退出其窗口)

1.3 什么是程序设计

在下面几章中,读者会知道,用 Visual Basic 编写程序包括编写指令,而该指令告知 Visual Basic 应执行哪些步骤。程序员试图用若干名字泛指这样的指令——语句、指令、命令和最常用的代码。在本章中,我们通常称它们为命令。

读者也许见过“程序设计语言”这个术语。目前,有许多程序设计语言,如 Basic, C, Pascal 和汇编语言,用这些语言就可以编写程序。总之,这些语言就像不同的会话语言,因为每种语言都有其自己的语法和一组词汇,而不同的语言都有其强点和弱点。许多专业程序员(包括在微软的程序员)都用 C 语言编写其程序,因为 C 语言功能强且具灵活性。不过, C 语言很难掌握,且用 C 语言编写 Windows 程序工作量很大。我们使用过所有计算机语言, Visual Basic 最易掌握和使用。

在本章中,通过使用 Visual Basic 的 Debug 窗口,读者将学会 Basic 语言中的三条命令。这一窗口允许读者输入命令,之后正常运行。运行命令的简单含义是,告知 Visual Basic 实际执行编程人员要它跟着执行的步骤。

1.4 使用 Debug 窗口

在启动 Visual Basic 时,读者也许已注意到, Visual Basic 使用了一定量的窗口——确切地说是 5 个窗口。现在,我们不理会这些窗口,因为在本章的余下部分中,编程人员完全在第六个窗口工作(在第三章,读者将使用其他窗口。在此期间,读者将学会如何在称为 Form1 的大窗口中画线。)

第六个窗口称为 Debug 窗口,用该窗口可以编写一程序。它之所以称为 Debug 窗口,是因为它允许用户输入命令(程序的一部分),而 Visual Basic 将立即执行之。如果还不够清楚,请不要着急,列举几个例子之后,读者会变得很清楚。

1.4.1 说明 Debug 窗口

在执行程序之前,用户必须知道如何打开 Debug 窗口。当其执行时,相当简单。其步骤如下:

- 按 F5,或下拉 Run 菜单并选择 Star。它告知 Visual Basic 从设计方式(后面用这种方式设计程序)切换为运行方式(实际上用这种方式运行程序)。
- 按 Ctrl+Break,或下拉 Run 菜单并选择 Break。这将 Debug 窗口带入前台,并使其成为活动窗口。

现在,我们就可看到 Debug 窗口了,如图 1.3 所示。

关键概念: Debug 窗口

Debug 窗口是测试新命令的有力工具,因为可以键入一条命令并及时看到结果。因为常常要使用 Debug 窗口,所以下面列举 Debug 窗口的执行步骤:

- 按 F5,或在 Run 菜单下选择 Star 项。

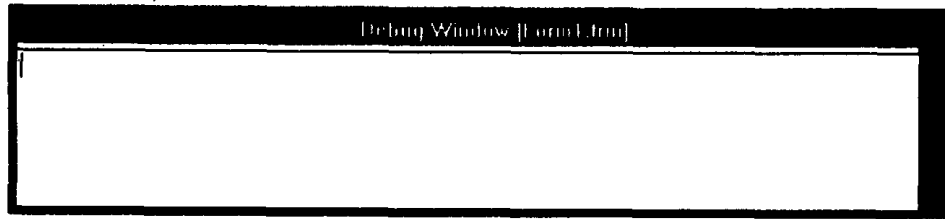


图 1.3 Debug 窗口。左上角的闪烁插入点将指明用户键入的正文将出现在何处

- 按 Ctrl+Break, 或在 Run 菜单下选择 Break 项。

1.4.2 在 Debug 窗口中键入内容

Debug 窗口与正文编辑器极为相似, 因为每键入一个字符, 字符便会出现在屏幕上, 且可使用鼠标器选择并编辑所键入的内容。但 Debug 窗口与正文编辑器有一个极为重要的区别: 当按 Enter 键时, Debug 窗口不仅将插入点移至下一行, 它还试图运行所键入的内容。

为了更为清楚地理解这一点, 用户可以键入一条非常简单的 Basic 命令—— Beep, 该命令使机器发出蜂鸣声。现在, 试键入该命令。只是键入 beep 并按 Enter 键就可以了(见图 1.4)。插入点移至下一行而机器发出蜂鸣音。

那已经是多少年以前了, 但每当我们回忆起第一次能这样输入命令时, 就感到极为兴奋。更为有趣的是, 我们可以在机器中键入一个字, 并用一个动作做为回答。在掌握这一技术之后, 程序设计就变得极为令人陶醉了。

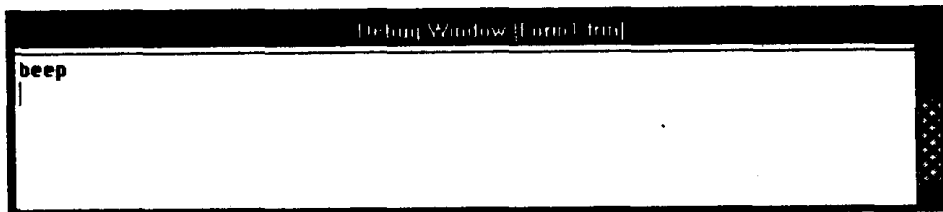


图 1.4 输入 Beep 命令后的 Debug 窗口, 它告知 Visual Basic 产生蜂鸣声

参考: Beep 命令

Beep 命令告知 Visual Basic 产生蜂鸣声。

由于用 Debug 窗口侧重于测试新命令, 所以让我们用几分钟时间开发这一窗口的某些功能。首先, 按 Backspace 键, 它使插入点移到上一行。如果再按退格键, 则会删除 Beep 命令中的字母 p。再键入 p, 之后按 Enter 键, Visual Basic 再次发出蜂鸣声。

现在, 让我们做进一步讲解。在两个 e 之间单击鼠标器左按钮, 此时, 插入点为:

```
be | ep
```

按 Enter 键。注意, Visual Basic 不是像人们所希望的那样将 ep 移至下一行, 而是再次发出蜂鸣声。换句话说, 在 Debug 窗口中按 Enter 键, 是告知 Visual Basic 在含有插入点的那一行运行命令。

在我们继续往下阅读之前,让我们开发 Debug 窗口的最后一个功能,编辑前面的命令。例如,当键入 Beep 命令时,键入有误,如将 Beep 键入为 Beeo。当按 Enter 键时,Visual Basic 将显示一个警告框,让用户知道录入有误,如图 1.5 所示。此时,该信息也许看似莫名其妙。结果,它实际上并不是这样莫名其妙。实际上,它表示 Visual Basic 未能找到与键入的名字相匹配的命令。

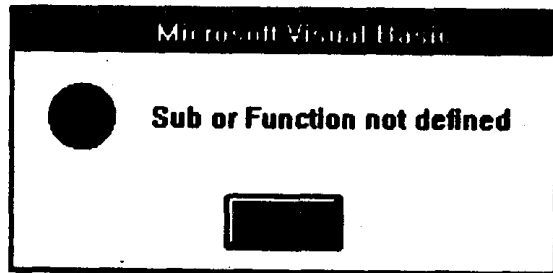


图 1.5 如果错误地键入一条命令,如将 Beep 录入为 Beeo,读者将看到这样的警告框。实际上,它表示“嘿?”

可以按 Enter 键,退出警告框,之后用鼠标器或用光标键编辑键入的内容。在执行完修改操作后,按 Enter 键,即可运行正确的命令。

1.5 离开运行方式

我们要学习的下一条命令称为 End 命令,它告知 Visual Basic 终止运行程序。换句话说,End 命令告知 Visual Basic 从运行方式切换到设计方式(不论何时,都可以看到 Debug 窗口)。

参考:End 命令

End 命令告知 Visual Basic 已运行完程序。当 Visual Basic 运行该命令时,返回到设计方式。

现在,试输入该命令。在 Debug 窗口键入 End 命令,之后按 Enter 键。读者会注意到,此时窗口从屏幕上消失而 Visual Basic 回到设计方式。这样,可以通过查看屏幕顶端了解目前使用何种方式。在设计方式下,屏幕顶行将为:

```
Microsoft Visual Basic [ design ]
```

方括号中的字(该例中为[design])指出当前何种方式为活动的。如果再次进入运行方式(按 F5),读者会注意到方式改为[run]。而当按 Ctrl+Break 以展示 Debug 窗口时,它切换到[break]。当我们熟悉了如何使用 Debug 窗口时,我们会讨论这三条命令。

Visual Basic 方式

Visual Basic 有三种方式。在读者阅读本书期间,应会使用这三种方式。通过查看屏幕顶端方括号中的字可以告知当前哪种方式是活动的:Microsoft Basic [design]。

- [design] 当设计程序时出现；
[run] 当 Visual Basic 运行当前程序时出现；
[break] 当程序终止时出现，它允许使用 Debug 窗口。

此外，还可以使用下拉式菜单，而不用 End 命令退出程序：下拉 Run 菜单并选择 End 项。

我们已学习了两个简单的 Basic 命令：Beep 和 End，并掌握了这两条命令的每个操作。在下面一节，我们将讨论一条更为有趣的命令，该命令可以执行多项操作。在开发 Basic 语言以及编写程序时，编程人员将时常使用该命令。

1.6 Print 命令

在这一节，读者将了解 Print 命令，用它可以探讨 Basic 如何处理算术操作。

Print 名字的由来

名字 Print 有点误导。许多人都认为，就好像是将输出内容从计算机传送到打印机。换句话说，Print 命令将其传送到屏幕。那为什么叫 Print 而不叫 Display 呢？其原因完全是历史性的。

Basic 语言起初是由 Dartmouth 学院的两名教授 John G. Kemeny 和 Thomas E. Kurtz 于 1963 年和 1964 年间建立的。当时，人们使用大型计算机而不是个人计算机。这些计算机在带有空调的大房子里装满了非常昂贵的设备，要与这样的计算机通信，需使用电传式，而不是 CRT 屏幕。因此，每当程序将输出内容传送给用户时，它实际上打印在一张纸上，而不是显示在屏幕上。这就是为什么 Kemeny 和 Kurtz 选择 Print 这个名字而不选择 Display 表示这条命令的原因所在。

名字 Basic 是一个字首组字，表示 Beginner's All-purpose Symbolic Instruction Code，这个名字一般用全大写字母书写：BASIC。不过，在 Visual Basic 中，Microsoft 将拼写方式改为大小写混合拼写方式。

要确保 Debug 窗口可见，因为我们将使用该窗口掌握 Print 命令。我们要做的第一件事是让 Print 显示一个数字。如果键入 Print 10，并按 Enter 键，会看见如下信息：

```
Print 10
      10
      |
```

注意，在按 Enter 键后，10 才出现，之后，插入点移至下一行。

现在，我们再换一换花样，将两个数相加。键入 Print 11+23，按 Enter 键。注意，Print 执行算术操作，并“打印”答案：

```
print 11+23
        34
        |
```

1.7 Basic 命令的构成

到目前为止，我们已使用过非常简单的 Basic 命令了。Beep 和 End 命令是可键入的简