

21世纪高等学校教材

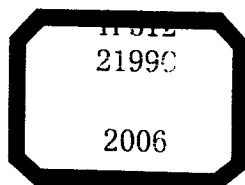
Visual Basic

程序设计 上机实验与习题解答

主编 冷金麟

上海交通大学出版社

21 世纪高等学校教材



Visual Basic 程序设计 上机实验与习题解答

主 编 冷金麟

副主编 于红光 周治钰

上海交通大学出版社

内 容 提 要

本书是与《Visual Basic 程序设计教程》配套的上机实验指导教材。全书分 4 章：第 1 章为上机指导；第 2 章为实验；第 3 章为习题；第 4 章为习题解答。

本书内容丰富，综合性强，同时对各章节的知识点加以总结，有利于学生知识的掌握和实践能力的提高。上机指导部分有助于学习者掌握上机预备知识，了解系统帮助的应用、应用程序的调试和错误处理以及应用程序安装盘的制作。实验部分针对各章节内容，重点归纳，综合案例实验实现过程分析，由浅入深、循序渐进地讲解了 Visual Basic 的基本应用程序的编写方法。习题的内容按简答题、选择题、填空题和程序设计题进行分类，提供了一些典型习题，帮助学生掌握所学的知识点，培养学生设计、编写 Visual Basic 程序的能力。习题解答部分包含习题、尤其是编程题的详解，方便学习者练习后对照检查。对于参加各种计算机考试的读者来说，该部分是具有实用性、针对性的辅导材料。

本书虽作为与《Visual Basic 程序设计教程》配套的上机实验指导教材，但其内容相对独立，可以与其他类似教材配套使用。

本套教材另外配有多媒体课件(PPT 格式)、所有案例的源程序与自动考试系统(适用于局域网，实现了理论知识和实际操作技能的全部自动化考核)。联系邮箱为：baiwen_sjtu@126.com

图书在版编目 (C I P) 数据

Visual Basic 程序设计上机实验与习题解答 / 冷金麟
主编. —上海: 上海交通大学出版社, 2006
ISBN 7-313-04618-9

I .V . . . II .冷 . . . III .BASICA 语言 - 程序设计 - 自学参考资料 IV .TP312

中国版本图书馆CIP数据核字 (2006) 第133856号

Visual Basic 程序设计

上机实验与习题解答

冷金麟 主编

上海交通大学出版社出版发行

(上海市番禺路 877 号 邮政编码 200030)

电话: 64071208 出版人: 张天蔚

常熟市文化印刷有限公司印刷 全国新华书店经销

开本: 787mm × 1092mm 1/16 印张: 12.75 字数: 307 千字

2006 年 11 月第 1 版 2006 年 11 月第 1 次印刷

ISBN7-313-04618-9/TP·665 定价: 19.00 元

版权所有 侵权必究

21 世纪高等学校教材

编审委员会

顾 问： 韩正之

执行主任： 百 文

副 主 任：	胡敬群	曹天守	靳全勤	张华隆	蒋凤瑛
	冯 颖	普杰信	程全洲	潘群娜	杨裕根
	徐祖茂	张红梅	宓一鸣	姜献峰	李 敏
	李湘梅	闫洪亮	陈树平	包奇金宝	刘克成
	白丽媛	戴 兵	张占山		

前 言

Visual Basic(简称VB)程序设计语言是目前最适合初级编程者学习使用的计算机高级语言之一。VB程序设计语言为用户提供了可视化的面向对象与事件驱动的程序设计集成环境,使得程序设计变得快捷、方便,简单易学,功能强大,应用广泛。近年来很多高校将VB作为学生学习程序设计的入门语言。

本书是与《Visual Basic 程序设计教程》配套的教学用书,从巩固学生的理论知识和提高学生的上机操作能力入手,按配套教程的章节和难易程度循序渐进地编写。全书分为4章:第1章为上机指导,讲述如何利用VB系统提供的在线帮助功能,认识VB系统,掌握系统功能的应用,同时还讲述在程序开发过程中重要的程序调试方法和技巧以及如何在VB系统的集成环境中制作、发布应用程序。第2章为实验,根据课程要求掌握的知识点提出实验的目的和要求,每个实验的实例都是精心编排、设计的,并给出详细的解题分析和操作步骤,以培养学生分析问题和解决问题的能力。第3章为习题,按配套教程的章节选编内容,力图通过习题的练习,使学生重温教材所讲述的理论与方法,加强学生对程序设计方法及算法分析技能的训练,培养学生创造性思考问题的能力。第4章为习题解答,对程序设计类习题的求解方法和步骤做出详细的讲解,以方便学生练习后对照检查。

本书的第1章由冷金麟编写,参加第2、3、4章编写的人员分别是李明(实验1、习题1及解答)、张桂容(实验2、习题2及解答)、宣善立(实验3、实验6、习题3及解答、习题6及解答)、冷金麟(实验4、实验8、习题4及解答、习题8及解答)、于红光(实验5、实验7、习题5及解答、习题7及解答)、周治钰(实验9、习题9及解答)、王杰和张继生(实验10、习题10及解答)。全书由冷金麟任主编并完成统编定稿,于红光、周治钰任副主编。

由于时间仓促,加之水平有限,书中难免有疏漏和不足之处,恳请广大读者和专家指正。

编 者
2006年9月

目 录

第 1 章 上机指导	1
1.1 使用联机帮助系统	1
1.2 应用程序调试及错误处理	3
1.3 制作应用程序的安装盘	12
第 2 章 实验	17
实验 1 Visual Basic 入门	17
实验 2 Visual Basic 编程基础	23
实验 3 Visual Basic 程序设计	31
实验 4 常用控件的使用	38
实验 5 数组	49
实验 6 过程	57
实验 7 用户界面的设计	66
实验 8 键盘、鼠标与绘图	72
实验 9 文件	84
实验 10 数据库的简单操作	97
第 3 章 习题	106
习题 1	106
习题 2	108
习题 3	112
习题 4	115
习题 5	121
习题 6	126
习题 7	130
习题 8	132
习题 9	137
习题 10	140

第 4 章 习题解答	143
习题 1	143
习题 2	148
习题 3	150
习题 4	158
习题 5	166
习题 6	170
习题 7	178
习题 8	182
习题 9	189
习题 10	192
参考文献	193

第 1 章 上机指导

Visual Basic(以下简称 VB)是 Microsoft 公司推出的可视化编程工具 Visual Studio 的众组件之一,是目前世界上使用最广泛的程序开发工具。VB 应用程序的开发以对象为基础,运用事件触发机制实现对 Windows 操作系统的事件响应。在 VB 提供的可视化的开发环境中,用户可以像搭积木一样构建出程序的界面,且 VB 还提供了丰富的控件,来帮助用户实现程序的功能。对于没有学习过任何一种计算机程序设计语言的人而言,VB 的语法是最容易被初学者所接收的。

1.1 使用联机帮助系统

VB 为用户提供了功能强大的联机帮助系统,以便用户更好地熟悉 VB 的编程环境、方法及开发界面的使用方法,掌握 VB 语言的基本语法,掌握 VB 各种控件的使用方法,并能将它们灵活运用到应用程序中,开发出具有一定功能的 VB 程序。VB 联机帮助系统详细说明了如何建立一个 VB 程序,如何编辑、编译、调试、运行、发布应用程序等。学会使用该系统,用户可以方便地解决在 VB 使用过程中遇到的大量问题。

从 Visual Studio 6.0 开始,所有的帮助文件都采用 MSDN(Microsoft Software Developer Network)文档的帮助方式。在 VB6.0 安装完成时,系统会提示用户安装 MSDN Library。MSDN Library 包含了超过 1GB 的编程技术信息,包括示例代码、文档、技术文章、Microsoft 开发人员知识库以及在使用该公司的技术来开发或解决方案时所需要的其他资料,内容非常详细全面。最新版的 MSDN 可以从 <http://www.microsoft.com/china/msdn> 上免费获取。此外,用户也可以使用 VB 的联机链接方式访问 Internet 上的相关网站获取更多信息。

1.1.1 获取全面的帮助——MSDN Library 查阅器

在 VB 操作过程中,用户若想查看关于 VB 的全面帮助信息,需通过 MSDN Library 查阅器打开 MSDN 帮助文档。启动 MSDN Library 查阅器有 3 种方法:

(1) 在 Windows 操作系统下的“开始”菜单的“程序”子菜单中,选择“Microsoft Developer Network”子菜单,单击“MSDN Library Visual Studio 6.0(CHS)”。

(2) 在 VB6.0 工作环境中,选择“帮助”菜单的“内容”、“索引”、“搜索”命令之一。

(3) 在 VB6.0 代码窗口设计时,直接按 F1 键。

打开后的 MSDN Library 查阅器的窗口如图 1-1 所示。

MSDN Library 查阅器窗口分为左、右两个显示区域,左边区域有 4 个选项卡,分别为“目录”、“索引”、“搜索”和“书签”。通过这 4 个选项卡可以用不同的方式定位和显示帮助文档。右边区域用于显示相应选项卡的内容。

(1) “目录”选项卡。单击图 1-1 中的“目录”选项卡可以目录树结构浏览各个标题。该目录包含了 MSDN Library 查阅器中所有可用信息的列表。单击工具栏上的“定位”按钮,

目录将会与用户所浏览的主题保持同步。单击树结构中的“+”或“-”图标可分别打开和关闭其中的帮助主题。

(2) “索引”选项卡。通过输入一个与所需信息有关的关键字或滚动翻阅整个列表可查找关键字。单击选中关键字后再单击“显示”按钮，或在列表中直接双击要查看的项目的关键字，窗口右侧将显示出相关的信息。

(3) “搜索”选项卡。通过“搜索”选项卡可查找包含在某个主题中的所有词或短语，用户可在输入栏中输入所要查找的词或短语，也可输入通配符表达式、布尔操作符、前一次搜索结果的列表、相似字匹配、主题的标题等来优化搜索。

(4) “书签”选项卡。在“书签”选项卡中，可以为需经常访问的信息创建书签列表，也可以通过已存在的书签直接访问相关信息。



图 1-1 MSDN Library 查阅器窗口

当用户在进行 VB 操作过程中需要获得窗口、控件、对象、属性、关键词或错误信息的帮助时，可选定要帮助的内容，然后直接按 F1 功能键，VB 系统便会根据用户所正在做的操作进行分析，打开 MSDN Library 查阅器直接定位显示相关的帮助信息。

1.1.2 获取最新的帮助——Web 上的 Microsoft

除了使用 MSDN Library 查阅器帮助方式外，用户还可以从 Internet 上获得 VB 的更多信息。在 VB6.0 中，用户可以在“帮助”菜单中选择“Web 上的 Microsoft”菜单，从弹出的子菜单中选择合适的选项(如图 1-2 所示)，进入 Microsoft 公司的主页，从网上查看或下载最新的帮助信息。

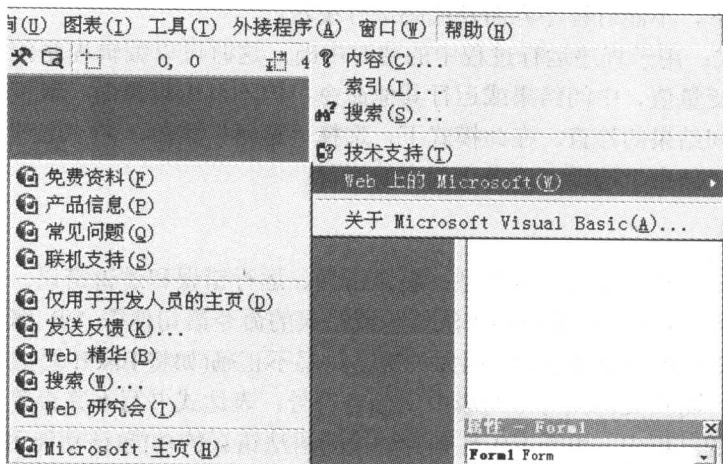


图 1-2 获得网上信息

1.1.3 获取深入的帮助——样例应用程序

为了帮助用户深入地学习和掌握 VB 中某些对象的使用，MSDN Library 提供了上百个 VB 的实例工程，缺省安装在“...\\Program Files\\Microsoft Visual Studio\\MSDN98\\98VS\\2052\\Samples\\VB98\\”子目录中。用户可在 VB 窗口中通过“文件”菜单里的“打开工程”命令，打开所需工程，观察其运行过程，分析事件代码，学习编程方法及技巧。

1.2 应用程序调试及错误处理

与面向过程的结构程序设计相比，VB 提供的面向对象的可视化开发环境，提高了程序设计开发的效率和质量。但是，无论程序员的编程技术多么高超，由于程序设计必须由程序员手工操作来完成，免不了会出现这样或那样的失误，因此，程序中也必然会出现这样或那样的错误。应用程序调试的目的就是为了尽量减少程序中存在的错误，以提高程序设计的质量。

1.2.1 应用程序调试

为了方便用户快速、准确地查找和修改错误，VB 系统提供一组交互式、有效的调试工具和手段。应用程序的调试就是在程序设计过程中不断地发现并改正错误。这是每一个程序员都必须熟练掌握的。

1. VB 的工作模式

VB 是一个集编辑、编译与运行于一体的集成开发系统。在程序设计过程中其工作状态可分为 3 种模式：设计模式、运行模式和中断模式。为了调试程序，用户必须知道系统当前所处的工作模式及其能实施的相关操作。

(1) 设计模式：用于用户基本界面的设计、相关控件的属性设置和程序代码编制。

(2) 运行模式：用于在编辑代码过程中编译执行应用程序，以观察运行效果或进行代码

调试。在该模式下，不能对程序界面及代码进行编辑。

(3) 中断模式：用于程序运行过程中的暂时中断，这时可以编辑程序代码，并可在“立即”窗口中显示变量值、中间结果或运行其他命令，但不可编辑界面。该模式主要用作程序代码的调试和中间结果的检查。在此模式下，选择“运行”菜单下的“继续”选项可以继续运行程序，选择“结束”选项可以中止程序的运行。

2. 程序错误

程序错误基本上可以分为 3 种类型：编译错误、运行错误和逻辑错误。

(1) 编译错误。由于使用错误的语法结构或错误的命令语句使得 VB 编译器无法对代码进行编译，这类错误称为编译错误。例如关键字拼写不正确(如将 ElseIf 写成 Else If)，遗漏标点符号、关键字，将英文标点符号写成中文标点符号，表达式书写不完善等。语法错误通常发生在键入代码的过程中。由于 VB 具有自动检测语法错误的功能(该功能可在“选项”对话框的“编辑器”选项卡中设置)，一旦检查出语法错误，会立即提示用户纠正。例如在图 1-3 中键入代码时误将“ElseIf”写成“Else If”，当光标离开该行时，系统立刻弹出编译错误提示对话框，显示该处代码有错误。此时单击“确定”按钮后，可对该处的错误代码进行纠正。

如果程序中存在不属于语法错误的错误代码，在键入代码时不会被语法检测发现，但在随后的程序运行时，系统在将程序编译成可执行文件时会提示错误。例如在图 1-4 中，在系统要求变量声明的情况下使用未定义变量，程序运行时 VB 将停止编译，并回到有错误的代码窗口，弹出错误提示对话框。此时，单击“确定”按钮后，可在“中断”模式下对代码窗口中的错误代码进行修改。

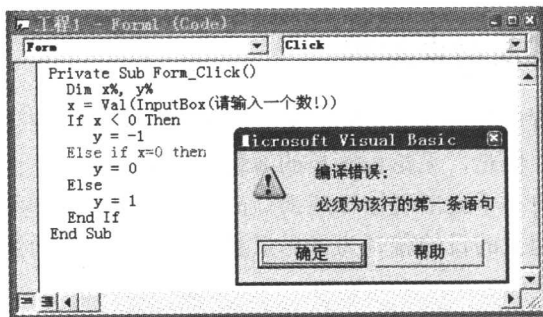


图 1-3 编译错误事例 1

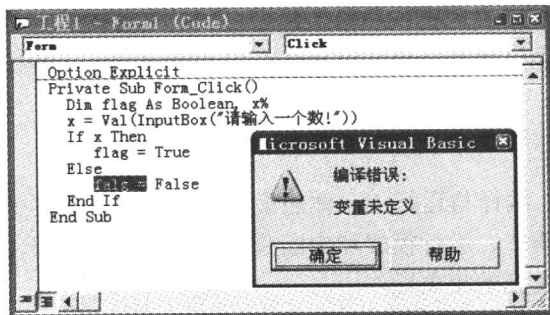


图 1-4 编译错误事例 2

(2) 运行错误。如果所键入代码的语法结构及程序的编译过程都正确无误,但在随后的运行过程中却发生错误,这通常是由程序在运行过程中执行了非法操作,或因某些操作失败而引起的。此类错误称为运行错误。例如,赋值语句的数据类型不匹配、试图打开一个不存在的文件、数组下标越界、磁盘存储空间不足等。在代码 1-1 中,属性 `FontSize` 的类型为整型,当对其赋值的类型为字符型时,系统运行将会显示如图 1-5 所示的出错信息。单击“调试”按钮,进入“中断”模式,光标停留在引起出错的语句上。此时可修改错误代码。

代码 1-1

```
Option Explicit
Private Sub Form_Load()
    Me.ForeColor = vbRed
    Me.BackColor = vbWhite
    Me.Caption = "测试窗体"
    Me.FontName = "隶书"
    Me.FontSize = "三号"
    Print "程序测试"
End Sub
```

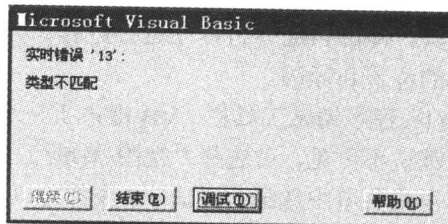


图 1-5 运行错误提示对话框

(3) 逻辑错误。程序从设计到编译运行,整个过程没有出现任何错误的提示,但运行后却得不到预期的结果,这类错误称为逻辑错误。例如运算符使用不当,语句的次序不正确,循环语句的条件值或初值、终值、步长设置不正确等等。由于这类错误不会导致错误信息的提示,因此错误较难排除,需要程序员仔细地阅读分析程序,并借助相应的调试工具才能查出原因,进行纠正。

例如图 1-6 是一个常见身份验证界面,运行时要求输入的用户名不能为纯数字,口令不能为空。根据程序要求,在两个文本框的 `LostFocus` 事件中分别编写了程序代码,见代码 1-2。

代码 1-2

```
Option Explicit
Private Sub Text1_LostFocus()
    If IsNumeric(Text1) Then
        MsgBox "用户名不能用纯数字!!"
        Text1.SetFocus '将操作焦点设置到text1上
    End If
End Sub
Private Sub Text2_LostFocus()
    If Len(Text2) = 0 Then
        MsgBox "请输入口令!"
        Text2.SetFocus '将操作焦点设置到Text2上
    End If
End Sub
```

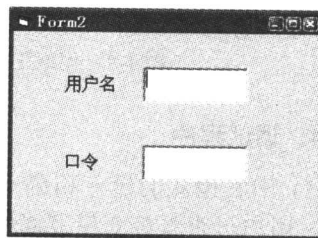


图 1-6 身份验证

这两段程序代码既没有编译错误,也未发生运行错误,但程序运行后可能会出现了死循环,具体过程是:当在用户名文本框中输入纯数字(如 12345)并按 `Tab` 键,将焦点移向口令文本框后,将触发用户名文本框 `Text1` 控件的 `LostFocus` 事件,由于输入的是纯数字,系统将在屏幕上弹出信息对话框,显示提示信息“用户名不能用纯数字!!”,并且通过指令 `Text1.SetFocus` 将操作焦点再设置到 `Text1` 上。当通过此指令将焦点由口令文本框 `Text2` 移到 `Text1` 上后,又将触发 `Text2` 控件的 `LostFocus` 事件,由于此时尚未在 `Text2` 中输入任何内容,因此, `Len(Text2)=0` 条件成立,系统又将在屏幕上显示出“请输入口令!”提示信息,并执行指令 `Text2.SetFocus`,将操作焦点由 `Text1` 再移到 `Text2` 上。当焦点从 `Text1` 移到 `Text2` 上后又再次

触发 Text1 控件的 LostFocus 事件。此时, Text1 中存放的仍是一开始输入的纯数字 12345, 因此, 系统又会执行指令 Text1.SetFocus, 将操作焦点从 Text2 移回到 Text1 上, 此指令的执行又将触发 Text2 控件的 LostFocus 事件……。由此, 系统程序运行进入了死循环状态。这种程序错误即为在逻辑上次序混乱的错误。

3. VB 的调试工具

程序的调试就是定位和修改那些使程序不能正确运行的错误。VB 提供了功能强大的调试工具, 能够帮助分析程序运行过程, 分析变量和属性值的变化, 可以便捷有效地查找错误产生的位置和原因。

(1) 程序调试工具栏。VB 提供了一个专用的程序调试工具栏, 在集成开发环境中, 该工具栏默认不可见, 可选择“视图/菜单/工具栏/调试”命令, 或在任何工具栏上单击鼠标右键, 在弹出式菜单中选择“调试”命令都可以打开调试工具栏(如图 1-7 所示)。利用该工具栏提供的按钮运行要测试的程序、中断程序的运行、在程序中设置断点、监视变量、单步调试和过程跟踪等以查找并排除代码中存在的逻辑错误。

(2) 调试菜单。除了通过打开调试工具栏可以进行调试以外, VB 还提供了“调试”菜单(如图 1-8 所示)和“运行”菜单。在菜单中也包含启动、中断、结束等功能命令。

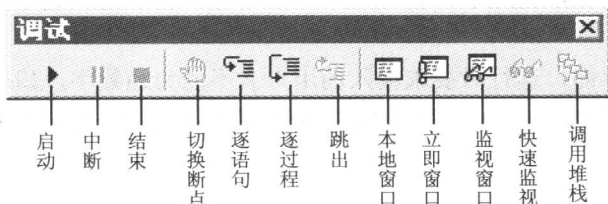


图 1-7 调试工具栏



图 1-8 “调试”菜单

4. 调试程序

(1) 中断模式的进入和退出。使用调试工具调试程序通常在中断模式下进行, 在中断状态下, 用户可查看各变量及属性的当前值, 观察界面状况, 从而了解程序执行是否正常。并且可以修改程序代码、修改变量及属性值等。

在程序调试时, 使系统进入中断模式的方法有以下几种:

- ① 程序运行时发生错误, 被系统检测到而中断。
- ② 程序运行中, 按组合键“Ctrl+Break”, 也可单击调试工具栏中“中断”按钮, 或选择“运行/中断”菜单项, 就会产生中断。
- ③ 在程序代码中设置断点, 当程序运行到断点处就会产生中断。
- ④ 采用逐语句或逐过程运行, 每执行完一行语句或一个过程后就会产生中断。
- ⑤ 在程序代码中使用 Stop 语句, 当执行到 Stop 语句时, 也会产生中断。

在中断模式下, 最便捷的查看程序中变量或属性的方法是将鼠标指针停留在要查看的变量上, 系统就会在随后弹出的一个小方框中显示出指针所指的变量或属性的当前值。

当在中断模式下调整完毕后, 退出中断模式的方法有:

① 如果要退出并继续运行程序，则可选择“运行/继续”菜单项，或单击调试工具栏中“继续”按钮。

② 如要结束运行，则可选择“运行/结束”菜单项，或单击调试工具栏中“结束”按钮。

(2) 控制程序的运行。在调试过程中，利用 VB 提供的调试工具可以方便地控制程序的运行。

① 启动。通过选择“运行/启动”菜单项，或单击调试工具栏中“启动”按钮便可启动程序运行。通常程序从头开始运行，如果设置了断点，程序就将运行到断点处停止，否则就运行到程序结束。如果在“工程属性”对话框中设置了“启动对象”，则工程从“启动对象”开始运行。

② 逐语句运行。逐语句运行即单步运行，每次只执行一条语句，之后运行中断，因此可逐个语句地检查每条语句的执行状况。当程序执行到过程调用语句时，逐语句将进入到被调过程的开始语句继续逐条运行。

按 F8 快捷键或单击调试工具栏中“逐语句”按钮，或选择“调试/逐语句”菜单项都可以单步运行。在代码编辑器窗口中，执行的语句前面有箭头和彩色背景，如图 1-9 所示。

③ 逐过程运行。当程序运行到调用过程时，逐过程运行可将整个被调用过程作为整体来执行。在程序调试过程中，当确认某些过程不存在错误时，使用逐过程运行可以不必对该过程进行逐语句运行，使运行更高效。

单击调试工具栏中的“逐过程”按钮，或按 Shift+F8 键，或选择“调试/逐过程”菜单项均可实现逐过程运行。如图 1-10 所示的程序运行过程，在程序运行“Swap1 a, b”语句时，使用逐过程运行，程序可以直接运行下一行输出语句，而不需进入 Swap1 子过程。

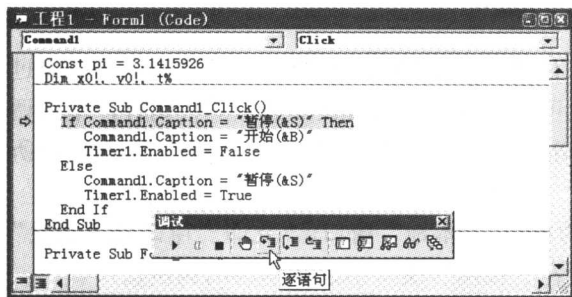


图 1-9 逐语句运行

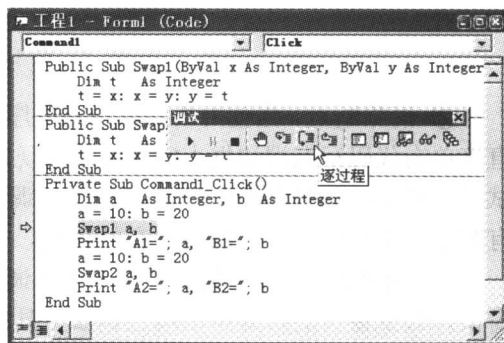


图 1-10 逐过程运行

④ 从过程中跳出。当程序运行进入被调用子过程后，若希望跳出该过程，返回到调用程序中的下一句语句时，可单击调试工具栏中“跳出”按钮或选择“调试/跳出”菜单项，从子过程中跳出。

⑤ 运行到光标处。选择“调试/运行到光标处”菜单项，可以让程序直接运行到光标所在位置停止以观察程序运行状况，而不需要通过逐语句执行每一行。实施的方法为：先将光标移到代码窗口中问题可能发生的代码行，然后选择“调试/运行到光标处”菜单项或按 Ctrl+F8 组合键，就可实现运行到光标处。

⑥ 设置下一条要执行的语句。在程序调试过程中，若希望程序运行只执行想要执行的代

码行, 跳过那些不想执行的代码行, 可通过设置下一条要执行的语句实现, 具体实施方法是在中断模式下, 先单击想执行的代码行, 再选择“调试/设置下一条要执行的语句”菜单项或按 **Ctrl+F9** 组合键, 然后按 **F5** 键或选择“运行/继续”菜单项来恢复执行。

⑦ 结束程序。若希望在程序运行过程中立即停止程序运行, 并返回到设计状态, 可直接单击调试工具栏中“结束”按钮或“运行/结束”菜单项即可结束程序运行。

(3) 断点的设置。使用断点是调试的重要手段, 断点使程序在指定的代码处自动停止执行, 并进入中断模式。断点通常设置在程序代码中能反映程序执行状况的关键代码行。例如, 可以在循环体中设置断点, 以了解每次循环中变量值的变化情况。

① 设置断点。在设计模式或中断模式下, 在代码编辑器窗口中单击将要设置断点的代码行左端的边框位置, 或将光标停放在该代码行上, 单击调试工具栏中“切换断点”按钮或选择“调试/切换断点”菜单项, 要设置断点的代码行字体被加粗且反白显示, 并在代码行左端边框上出现圆点。图 1-11 显示了在代码窗口中设置了两个断点, 当程序运行到断点处时暂停运行, 代码行左端的断点上有一水平向右的箭头表明程序在此代码行暂停运行。当鼠标指针指向需观察的变量或属性时, 就会在该变量或属性的下方显示出其值。

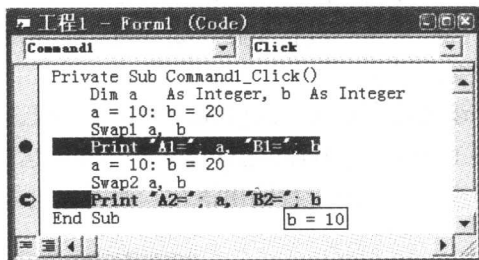


图 1-11 设置断点

② 清除断点。当使用断点调试完程序后, 应清除程序代码中的断点。清除断点可以通过直接单击代码行左端边框上的圆点来实现, 也可在光标停留在被设置断点的代码行, 单击调试工具栏上“切换断点”按钮或选择“调试/切换断点”菜单项清除。若选择“调试/清除所有断点”菜单项可一次清除所有的断点。

(4) 调试窗口。VB 提供了 3 个用于调试的窗口: “本地”窗口、“立即”窗口和“监视”窗口。调试窗口的打开可以通过调试工具栏或在“视图”菜单中选择相应的调试窗口。

① “本地”窗口。“本地”窗口可以显示当前过程中所有变量的值。“本地”窗口只能显示本过程中的变量, 其他过程的变量则无法看到。

图 1-12 所示为工程 1.Form1.Command1_Click 过程, 单击“Me”前面的“+”图标, 可以看到窗体的所有信息。

② “立即”窗口。“立即”窗口用于显示在程序运行过程中与当前过程有关的信息, 可以显示某个变量或属性值, 或执行单个过程或表达式。在进入中断模式后, 单击调试工具栏中“立即”窗口按钮显示“立即”窗口。

在立即窗口中可以实现以下功能:

a. 用 `Debug.Print` 方法输出信息。调试程序时可在程序代码中添加 `Debug.Print n` 等语句, 将变量或表达式的值输出到“立即”窗口中。程序调试完成后, 应将 `Debug.Print` 语句删除。

如图 1-13 所示, 在 fac()函数过程中使用了 Debug. Print 语句:

```
Debug.Print "fac="; fac, "n="; n
```

显示每次调用 fac()函数时, fac()函数值及 n 的变量值。

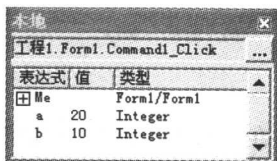


图 1-12 “本地”窗口

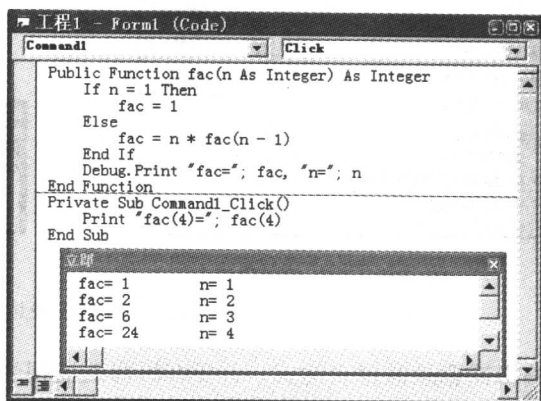


图 1-13 “立即”窗口显示反复调用函数的执行过程

b. 在设计时用来测试表达式。在设计模式下, 可以在“立即”窗口中使用 Print 方法或? 直接输出一些表达式的结果。

例如, 在“立即”窗口中输入:

```
? chr$(Int(rnd*26+65))
```

按 Enter 键立即显示一个随机产生的大写字母。

c. 直接从“立即”窗口打印。在中断模式下, 通过使用 Print 方法或? 可在“立即”窗口中检查变量或表达式的值。

例如, 在“立即”窗口中输入:

```
? i, j, i*j
```

d. 从“立即”窗口编辑变量或属性值。在中断模式下, 可在“立即”窗口中设置变量或属性的值。

例如, 当程序中断时, 在“立即”窗口中给属性重新赋值:

```
Text1.ForeColor=vbRed
```

当程序继续运行时, 文本框中字体的颜色就可变成红色。

e. 从“立即”窗口测试过程。从“立即”窗口可以通过指定参数值来调用过程, 以测试程序的正确性。

例如, 使用新的参数来计算调用函数 fac()的结果:

```
n=5
```

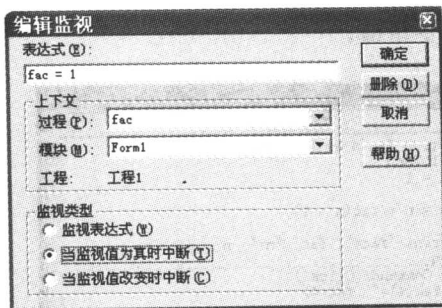
```
? fac(n)
```

将 n 作为 fac()函数的参数, 然后调用 fac()函数并显示返回值。

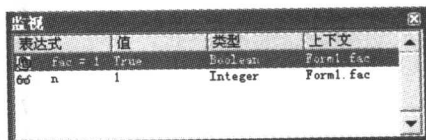
③ “监视”窗口。在程序运行过程中, 若需要了解多个表达式的值的变化, 可在中断模式下, 通过“监视”窗口获取, 下面是使用“监视”窗口监视表达式的步骤:

a. 添加监视表达式。选择“调试/添加监视”菜单项, 显示如图 1-14(a)所示的对话框。在“表达式”编辑框中输入表达式, 在“上下文”区域中的下拉组合框中分别选择表达式所

在的过程名和模块名, 则表达式就添加到“监视”窗口中, 如图 1-14(b)。



(a) 添加“监视”对话框



(b) “监视”窗口

图 1-14 添加监视表达式窗口

b. 编辑或删除监视表达式。若要编辑修改监视表达式, 可在“监视”窗口中选定要编辑的表达式, 选择“调试/编辑监视”菜单项或按右键在弹出的快捷菜单中选择“编辑监视”菜单项, 在弹出的编辑监视对话框中进行编辑。

c. 快速监视。在程序运行过程中, 若要监视未添加的表达式值, 可以采用快速监视的方法。在中断模式下, 在代码编辑器窗口中选定需要监视的表达式或属性, 再选择“调试/快速监视”菜单项或按 Shift+F9 组合键, 就可在弹出的对话框中查看相应的值, 如图 1-15 所示。

(5) 调用堆栈。调用堆栈用于显示一个调用的所有活动过程列表, 调用活动过程是指已经启动但还未执行完的过程。

在中断模式下, 单击调试工具栏中“调用堆栈”按钮或选择“调试/调用堆栈”菜单项, 就可出现“调用堆栈”的对话框, 如图 1-16 所示。

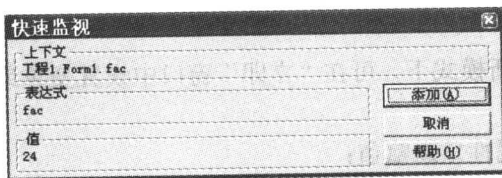


图 1-15 “快速监视”窗口

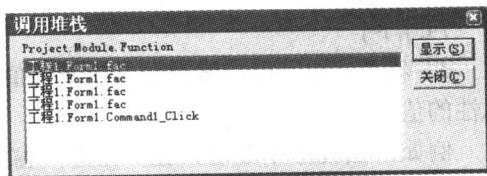


图 1-16 “调用堆栈”窗口

在对话框中由下往上, 依次显示的是调用过程、被调用过程、被调用过程……。如果要详细显示某一过程的语句, 可用鼠标选中该过程, 再单击“显示”按钮即可。

1.2.2 错误处理

程序运行时产生的任何错误都可能会导致程序运行的中止。尽管在程序编写完成后都要进行严格调试, 但这只能是尽可能多地减少错误, 而不可能彻底根除错误。此外, 在实际应用中还可能由于运行环境、资源使用等原因出现错误, 资源路径不正确、文件名输入有误, 磁盘驱动器出现故障等, 都可能会发生运行错误。因此, 在开始编程时就需要事先将错误考虑进去, 以便捕获错误、处理错误。

错误处理实际上就是通过程序在运行中捕获错误, 并进行适当地处理, 对于可能出错的