

Visual Basic 程序设计实验指导 与习题解答

陈达 吴劲芸 主编

VB CHENGXU SHEJI
SHIYAN ZHIDAO YU XITI JIEDA



湖北长江出版集团
湖北科学技术出版社

《Visual Basic 程序设计教程实验指导与习题解答》编委会

主 编	陈 达	吴劲芸		
主 审	赵 臻	吴长海		
副主编	刘 艳	彭 瑜	蒋厚亮	陈罗庚
编 委	邓文萍	邓贞嵘	胡 芳	解 丹
	李卫平	沈绍武	孙杨波	王 慧
	伍宁杰	夏 伟	曾洁玲	

内 容 简 介

本书是《Visual Basic 程序设计教程》一书配套的辅助教材。

全书共分4个部分。第1部分为上机指导,详细介绍了在上机过程中会遇到的一些问题及上机编写、调试程序的基本过程。第2部分为实验内容,根据教材各章讲述的相关内容,编排了11个综合实验题目,帮助读者掌握学习内容。第3部分是《Visual Basic 程序设计教程》的习题解答篇,按照教材各章习题编写的习题参考答案,大多数习题都给出了详细的解题分析过程。第4部分是综合本教程的内容要求并结合全国计算机等级考试的内容形式给出了几套综合测验题。

本书习题内容解答详细,针对性强;实验内容丰富,综合性强;并对各章节的知识点加以适当扩充,实验的难度及复杂性相对主教材例题有所提高,有利于学生知识的掌握和实践能力的提高。

本书可与《Visual Basic 程序设计教程》一书配套使用,也可作为全国计算机等级考试VB程序设计的习题参考用书,还可作为相关技术培训的教材以及程序设计初学者的自学用书。

前 言

Visual Basic 是 Microsoft 公司推出的一种面向对象的“可视化”Windows 应用程序开发工具,它在语法上继承了 Basic 和 Quick Basic 的优点,具有使用方便、简单易学等特点,且功能强大,与其他开发工具有丰富的接口。因此,深受广大用户的青睐,成为学习开发 Windows 应用程序首选的程序设计语言。

目前,许多高校非计算机专业都开设了 Visual Basic 程序设计课程,而很多非计算机专业人员也选择使用 Visual Basic 作为学习计算机程序设计的语言。

“Visual Basic 程序设计”是一门实践性很强的课程,我们根据多年从事计算机程序设计教学实践经验,编写了这本《Visual Basic 程序设计实验指导与习题解答》。按照 Visual Basic 程序设计的特点,采用“任务驱动”方式,精心设计每一个实验实例、实验内容。实践证明,这更能激发读者学习 Visual Basic 程序设计的兴趣,培养学生的实际编程能力。

全书包括 Visual Basic 上机基本指导、基础实验内容与指导、习题参考解答和部分综合测试题。其中 Visual Basic 上机基本指导部分主要介绍 Visual Basic 6.0 集成环境的使用与设置和 Visual Basic 程序设计的基本概念;基础实验与指导部分共有 11 个实验,是针对程序设计初学者而设计的,主要包括大学非计算机专业 Visual Basic 程序设计课程的教学实验内容。所有实验均具有较强的针对性和实践性,通过实验使读者掌握 Visual Basic 程序设计与调试方法,巩固所学知识,培养实际编程能力。

习题参考解答部分给出了《Visual Basic 程序设计教程》中各章习题的详细分析、解题思路及方法;综合测试练习题用于读者复习、巩固所学知识,测验自己是否真正掌握了相关知识。

本书可以作为各类高等院校、各类高职院校非计算机专业学生的“Visual Basic 程序设计”课程的实验教学用书,也可作为广大计算机爱好者学习 Visual Basic 程序设计语言的参考书。

由于作者水平有限,书中一些不当或错误谨请广大读者批评指正。

作者

2007 年 2 月

目 录

第1部分 上机指导	1
1.1 VB 简介	1
1.1.1 VB 功能特点	1
1.1.2 运行环境、安装和启动	1
1.1.3 集成开发环境	2
1.1.4 VB 集成开发环境中的各类窗口	2
1.2 VB 代码的一些约定	4
1.2.1 变量和对象的前缀	4
1.2.2 控件或窗体、模块对象的前缀	5
1.2.3 变量加额外的前缀	7
1.2.4 关于变量的使用	7
1.2.5 窗体布局	8
1.2.6 代码	9
1.3 应用程序调试	10
1.3.1 VB 的工作模式	11
1.3.2 程序错误种类	11
1.3.3 使用 VB 的调试工具	13
1.4 错误处理	25
1.4.1 错误的捕获及处理子程序	25
1.4.2 Error 函数和 Error 语句	27
1.5 制作 EXE 可执行文件	27
1.6 使用打包和展开向导	28
第2部分 实验内容	32
VB 实验一(设计简单的 VB 应用程序)	32
VB 实验二(设计简单的 VB 应用程序)	35
VB 实验三(Visual Basic 程序设计基础)	37
VB 实验四(顺序结构程序设计)	39
VB 实验五(选择及循环)	40
VB 实验六(常用控件)	53
VB 实验七(数组)	59
VB 实验八(过程及函数)	61
VB 实验九(菜单制作)	65
VB 实验十(对话框)	68
VB 实验十一(文件)	73

第 3 部分 习题选解	76
第 1 章 参考答案	76
第 2 章 参考答案	79
第 3 章 参考答案	82
第 4 章 参考答案	84
第 5 章 参考答案	89
第 6 章 参考答案	99
第 7 章 参考答案	114
第 8 章 参考答案	125
第 9 章 制作菜单答案	136
第 10 章 对话框程序设计答案	137
第 11 章 文件答案	138
第 4 部分 综合练习题	147
综合练习 1	147
综合练习 2	164
综合练习 3	184
综合练习题参考答案	202
附录	204
参考文献	208

第1部分 上机指导

1.1 VB 简介

1991 年 MS 公司推出的 Windows 应用程序开发工具——Visual Basic, 是基于 Basic 语言的可视化程序设计语言。它既继承了 Basic 所具有的程序设计语言简单易用的特点, 又采用了面向对象、事件驱动的编程机制, 用巧妙的方法把 Windows 编程的复杂性封装起来, 提供了所见即所得的可视界面设计方法。

1993 年推出的 VB3.0 版本初具规模。1995 年推出的 VB4.0 版本, 1997 年推出的 VB5.0 版本, 1998 年推出的 VB6.0 版本。

VB6.0 包括三个版本, 学习版、专业版、企业版。

1.1.1 VB 功能特点

- (1) 具有面向对象的可视化设计工具。
- (2) 事件驱动的编程机制。
- (3) 提供易学易用的应用程序集成开发环境。
- (4) 结构化的程序设计语言。
- (5) 强大的数据库功能。
- (6) ActiveX 技术。
- (7) 网络功能。
- (8) 多个应用程序向导。
- (9) 完备的 Help 联机帮助功能 (Microsoft Developer Network Library, MSDN)。

1.1.2 运行环境、安装和启动

1. 运行环境

系统基本要求:

80486 或更高的微处理器

不小于 50MB 硬盘空间

一个 CD-ROM 驱动器

至少 16MB RAM

VGA 或更高分辨率的监视器。

2. 软件要求

Microsoft Windows 2000 或 Microsoft Windows NT 4.0 及以上版本。

3. 安装

运行安装盘根目录上的 SETUP 程序, 按提示进行安装即可。

如果使用 Visual Studio 产品, 系统还会提示是否安装 MSDN。

4. 启动

- 双击桌面上的 Visual Basic 的快捷图标

- 开始——程序——Visual Basic 6.0

启动 VB 后,出现新建工程对话框,作用如下:

新建:建立新工程。

现存:选择和打开现有的工程。

最新:列出最近使用过的工程。

5. 退出

- 文件菜单中的“退出”

- Visual Basic 6.0 窗口的“关闭”按钮

- Alt + Q 组合键

1.1.3 集成开发环境

所谓集成开发环境,就是在一个公共开发环境中集成了设计、编辑、编译、调试、运行、帮助等诸多功能的系统平台。其中每个功能都是以一个独立的程序运行,并都有自己的界面。Visual Basic 6.0 的集成开发环境除了常用的标题栏、菜单栏、工具栏外,还包括一些特殊的窗口。

1. 标题栏

- 设计模式:可进行用户界面的设计和代码的编制。

- 运行模式:运行应用程序,不能编辑代码和界面。

- 中断模式(break):运行的程序暂时中断,可编辑代码,不能编辑界面。F5 键可继续运行。

2. 菜单栏

共有 13 个下拉菜单,用于显示所使用的 Visual Basic 的命令。

3. 工具栏

提供了菜单常用命令的快速方式。

4. 快捷菜单

显示使用对象的常用操作命令。

1.1.4 VB 集成开发环境中的各类窗口

1. 窗体(Form)窗口

窗体(图 1-1)是建立 VB 应用程序的主要部分,用户通过与窗体上的控制部件交互操作可得到相应运行结果。每个窗体必须有一个唯一的窗体名字,默认是 Form1、Form2 等。一个应用程序至少有一个窗体窗口,用户可在应用程序中设置多个窗体窗口。

2. 属性(Properties)窗口

属性窗口(图 1-2)中可以设置所有窗体或控件的属性,如颜色、字体、大小等。

属性窗口的组成如下:

对象列表框

属性显示排列方式有“字母”和“分类”两种

属性列表框

属性含义说明

3. 工程资源管理器(Project Explorer)窗口

工程资源管理器窗口(图 1-3)显示组成一个工程的所有文件,它的扩展名是 .VBP。

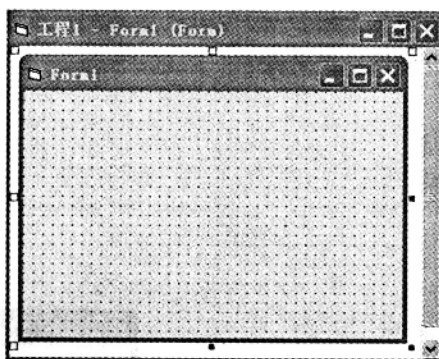


图 1-1 窗体窗口

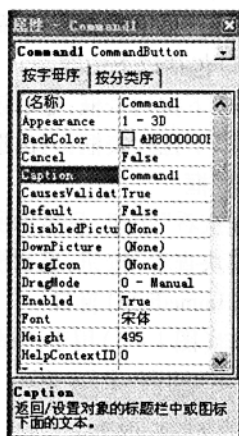


图 1-2 属性窗口

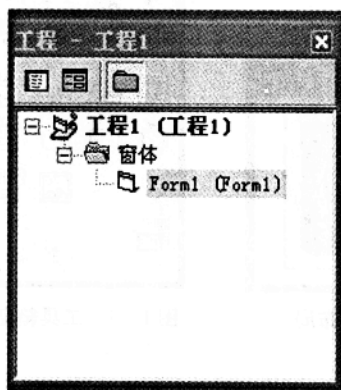


图 1-3 工程资源管理器窗口

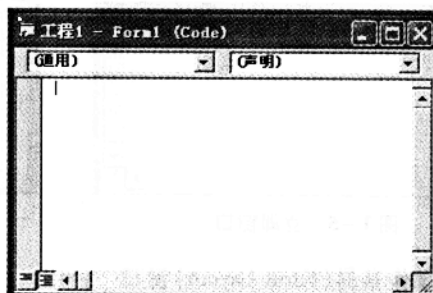


图 1-4 代码窗口

三个按钮分别是：

- 查看代码按钮
- 查看对象按钮
- 切换文件夹按钮

窗体文件(.frm 文件):存放窗体上使用的所有的控件对象和有关的属性、对象相应的事件过程、程序代码。

标准模块文件(.bas 文件):用户自定义的通过程,通用就是可以被应用程序各处调用的过程。

类模块文件(.cls 文件):可以用类模块建立自己的对象。类模块包含用户对象的属性和方法。

注意:列表框中括号左边的部分表示此工程、窗体、标准模块的名称(即属性中 Name 项的值,可在程序代码中使用);括号中的部分时报尊在磁盘的文件名。无扩展名时表示该文件还没存盘。

4. 代码(Code)窗口

代码窗口(图 1-4)专门用来进行程序设计的窗口,显示和编辑程序代码。打开代码窗口有三种方法:

- 选中一个窗体或标准模块后,单击“查看代码”按钮。
- 双击一个控件或窗体本身。

●从“视图”菜单中选择“代码窗口”项。

代码窗口包括以下内容。

- “对象”下拉列表框:显示该窗体中包含的对象名称
- “过程”下拉列表框:列出选中对象的所有事件过程名称
- “代码”框
- “过程查看”按钮:只显示所选的一个过程
- “全模块查看”按钮:显示模块中全部过程

5. 立即 (Immediate) 窗口

立即窗口(图 1-5)用于调试应用程序。

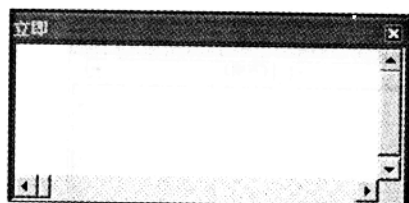


图 1-5 立即窗口

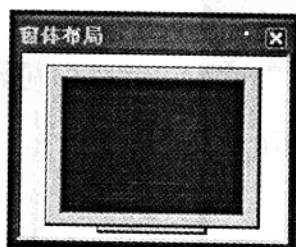


图 1-6 窗体布局



图 1-7 工具箱窗口

6. 窗体布局 (Form Layout) 窗口

窗体布局窗口(图 1-6)用于调整和显示窗体在屏幕中的位置。

7. 工具箱 (Toolbox) 窗口

在工具箱中存放 20 个标准控件,用于设计可视仪程序的外观(图 1-7)。

1.2 VB 代码的一些约定

1.2.1 变量和对象的前缀

程序中的变量遵循的表示法为“前缀 + 变量含义”,变量的含义为一个或多个英文单词,每个单词的第一个字母大写,不要用汉语拼音代替。变量前缀为三个小写字母示意其类型,所需添加的前缀遵循微软在 MSDN 中的建议,列表(表 1-1)如下。

表 1-1 基本数据类型的前缀

变量类型	前缀	示例
Boolean	bln	blnFound
Byte	byt	bytRasterData
Collection object	col	colWidgets
Currency	cur	curRevenue
Date (Time)	dtm	dtmStart

续表

变量类型	前缀	示例
Double	dbl	dblTolerance
Error	err	errOrderNum
Integer	int	intQuantity
Long	lng	lngDistance
Object	obj	objCurrent
Single	sng	sngAverage
String	str	strFName
User - defined type	udt	udtEmployee
Variant	vnt	vntChecksum

1.2.2 控件或窗体、模块对象的前缀

表 1-2 控件或窗体、模块对象的前缀

变量类型	前缀	示例
3D Panel	pnl	pnlGroup
ADO Data	ado	adoBiblio
Animated button	ani	aniMailBox
Check box	chk	chkReadOnly
Combo box, drop - down list box	cbo	cboEnglish
Command button	cmd	cmdExit
Common dialog	dlg	dlgFileOpen
Communications	com	comFax
Control (在过程中使用的类型未明的控件变量)	ctr	ctrCurrent
Data	dat	datBiblio
Data - bound combo box	dbcbo	dbcboLanguage
Data - bound grid	dbgrd	dbgrdQueryResult
Data - bound list box	dblst	dblstJobType
Data combo	dbc	dbcAuthor
Data grid	dgd	dgdTitles
Data list	dbl	dblPublisher

续表

变量类型	前缀	示例
Data repeater	drp	drpLocation
Date picker	dtp	dtpPublished
Directory list box	dir	dirSource
Drive list box	drv	drvTarget
File list box	fil	filSource
Flat scroll bar	fsb	fsbMove
Form	frm	frmEntry
Frame	fra	fraLanguage
Gauge	gau	gauStatus
Graph	gra	graRevenue
Grid	grd	grdPrices
Hierarchical flexgrid	flex	flexOrders
Horizontal scroll bar	hsb	hsbVolume
Image	img	imgIcon
Image combo	imgcbo	imgcboProduct
ImageList	ils	ilsAllIcons
Label	lbl	lblHelpMessage
Lightweight check box	lwchk	lwchkArchive
Lightweight combo box	lwco	lwcoGerman
Lightweight commandbutton	lwcmb	lwcmbRemove
Lightweight frame	lwfra	lwfraSaveOptions
Lightweight horizontal scroll bar	Lwhsb	lwhsbVolume
Lightweight list box	lwlst	lwlstCostCenters
Lightweight option button	lwopt	lwoptIncomeLevel
Lightweight text box	lwtxt	lwoptStreet
Lightweight vertical scroll bar	Lwvsb	lwvsbYear
Line	lin	linVertical
List box	lst	lstPolicyCodes
ListView	lvw	lvwHeadings
MAPI message	mpm	mpmSentMessage
MAPI session	mps	mpsSession
MCI	mci	mciVideo
Menu	mnu	mnuFileOpen

续表

变量类型	前缀	示例
Month view	mvw	mvwPeriod
MS Chart	ch	chSalesbyRegion
MS Flex grid	msg	msgClients
MS Tab	mst	mstFirst
OLE container	ole	oleWorksheet
Option button	opt	optGender
Picture box	pic	picVGA
Picture clip	clp	clpToolBar
ProgressBar	prg	prgLoadFile
Remote Data	rd	rdTitles
RichTextBox	rtf	rtfReport
Shape	shp	shpCircle
Slider	sld	sldScale
Spin	spn	spnPages
StatusBar	sta	staDateTime
SysInfo	sys	sysMonitor
TabStrip	tab	tabOptions
Text box	txt	txtLastName
Timer	tmr	tmrAlarm
ToolBar	tlb	tlbActions
TreeView	tre	treOrganization
UpDown	upd	updDirection
Vertical scroll bar	vsb	vsbRate

1.2.3 变量加额外的前缀

除对于用户使用 type 关键字定义的数据类型,在三个字母的此之外,还要对于一些不同的级别的变量加额外的前缀,举例如下(表 1-3)。

表 1-3 级别的变量加额外的前缀

级别	前缀	示例
全局变量	g	gstrUserName
模块级变量	m	mblnCalcInProgress
过程级变量	N	ndblVelocity

1.2.4 关于变量的使用

关于变量使用方面,建议如下:

(1) 变量要先声明再使用(在窗体代码的第一行加上 Option Explicit,来禁止未声明变量的调用,或者在菜单 Tools - > Options - > Editor 中选中 Require Variable Declaration 项)。

(2) 尽量使用 Long 型的变量来代替 Integer 类型,这样做可以减少一些数据溢出的错误,而且,在 Win32 平台上,CPU 处理 32 位的数据比 16 位的数据速度更快。

(3) 尽量少用 Variant 变量,尽可能地给出每一个变量明确的类型。

(4) 尽量不在 API 声明中使用 As Any,如果遇到其默认的参数为 As Any 的情况,则针对每一个所需的参数类型声明该 API 函数,例如 ReadFile 函数默认的声明如下。

```
Public Declare Function ReadFile Lib "kernel32" Alias "ReadFile" (ByVal hFile As Long, lpBuffer As Any, ByVal nNumberOfBytesToRead As Long, lpNumberOfBytesRead As Long, lpOverlapped As OVERLAPPED) As Long
```

小组成员须将其改为:

```
Public Declare Function ReadFileByt Lib "kernel32" Alias "ReadFile" (ByVal hFile As Long, lpBuffer As Byte, ByVal nNumberOfBytesToRead As Long, lpNumberOfBytesRead As Long, lpOverlapped As OVERLAPPED) As Long
```

```
Public Declare Function ReadFileInt Lib "kernel32" Alias "ReadFile" (ByVal hFile As Long, lpBuffer As Integer, ByVal nNumberOfBytesToRead As Long, lpNumberOfBytesRead As Long, lpOverlapped As OVERLAPPED) As Long
```

```
Public Declare Function ReadFileLng Lib "kernel32" Alias "ReadFile" (ByVal hFile As Long, lpBuffer As Long, ByVal nNumberOfBytesToRead As Long, lpNumberOfBytesRead As Long, lpOverlapped As OVERLAPPED) As Long
```

(5) 不要将类型转换的工作交给 VB 自动去做,而使用以下类型转换函数。

表 1-4 型转换函数

CBool(expression)

CByte(expression)

CCur(expression)

CDate(expression)

CDbl(expression)

CDec(expression)

CInt(expression)

CLng(expression)

CSng(expression)

CStr(expression)

CVar(expression)

1.2.5 窗体布局

窗体内各个控件的布局,建议如下:

(1) 所有与窗体上、下、左、右四条边界相邻的控件与窗体的边缘相距 120 缇。

(2) 各个控件之间的间距为:相关控件 60 缇,非相关控件 120 缇。

(3)按钮控件大小建议为高 300 缇,长 1200 缇,这也是操作系统默认按钮的大小。

(4)各个控件的字体建议为宋体五号字。

1.2.6 代码

1. 程序的启动对象

程序从 Main() 函数开始执行(选择菜单 View -> Project Explorer, 在工程窗口内右键点击当前项目, 选择菜单“... Properties” -> General, 在 Startup Object 下拉框中选择 sub main)

2. 代码缩进与间距

每个单位的代码缩进为一个 TAB, 非相关代码留一行空白的间隔, 例如:

```
Private Function GetMax(Byref lngArray() As Long) As Long
```

```
GetMax = 0
```

```
    Dim lngMax As Long
```

```
    Dim lngCount as long
```

```
    For lngCount = 0 to Ubound(lngArray)
```

```
        If lngArray (lngCount) > lngMax Then
```

```
            lngMax = lngArray(lngCount)
```

```
        End If
```

```
    Next
```

```
End Function
```

3. 注释

程序的注释越详尽、越仔细越好。

程序中声明的每一个变量, 能加上注释最好, 至少用来运算或者保存关键数据的变量必须加上注释。

对于每一段逻辑上实现某个基本功能的几句代码, 最好在其上一行加上简单注释。

对于每一个自定义的函数, 不管大小都必须加上注释, 注释的格式如下:

```
' * * * * *'
```

```
'函数所实现的功能
```

```
'函数的参数 1 的含义:XXXXX
```

```
'函数的参数 2 的含义:XXXXX
```

```
'.....
```

```
'函数返回值所代表的错误信息:XXXXX(函数尽可能声明为 Function, 不要声明为 Sub,
```

```
'函数的返回值为 0 表示执行成功, 为其它值表示执行失败)
```

```
'
```

```
' * * * * *'
```

```
Public (Private) Function ForExample(.....) As Long
```

```
'.....
```

```
End Function
```

4. 错误处理

在 Main() 函数或主窗体 Load 过程中添加如下代码:

```
' * * * * *'
```

```
'打开错误日志文件, 在 Form_Unload() 中关闭
```

```
'
```

```
'*****
```

```
Dim strExePath As String
```

```
If Right(App.Path, 1) = " " Then
```

```
    strExePath = App.Path
```

```
Else
```

```
    strExePath = App.Path & " "
```

```
End If
```

```
intErrLogFileHandle = FreeFile()
```

```
Open strExePath + "Err.log" For Append Shared As intErrLogFileHandle
```

在主窗体退出函数中添加如下代码:

```
'*****
```

```
'
```

关闭错误日志文件

```
'*****
```

```
Close #intErrLogFileHandle
```

在某个 Module 中添加:

```
'*****
```

```
'
```

写入错误日志

```
'
```

```
'*****
```

```
Public Sub WriteErrLogFile(ByVal strSub As String, ByVal strErr As String)
```

```
    Print #intErrLogFileHandle, Date, Time, strSub, strErr
```

```
End Sub
```

在每个过程或自定义的函数中,使用如下的系统错误捕获机制:

```
Public Function ForExample(.....) As Long
```

```
On Error Goto FuncError
```

```
    ForExample = True
```

```
    .....
```

```
    .....
```

```
Exit Function
```

```
FuncError:
```

```
    ForExample = False
```

WriteErrLogFile "ForExample", "错误号:" & Err.Number & ", 错误源:" & Err.Source & ", 错误描述:" & Err.Description

```
    MsgBox ".....", vbCritical
```

在此作内存资源回收工作

```
End Function
```

1.3 应用程序调试

在程序编写的过程中,出现错误是难免的,查找并修改错误的过程称为程序调试。

1.3.1 VB 的工作模式

在进行程序调试时,知道当前应用程序正处在何种工作模式下是非常重要的。VB 提供三种工作模式:设计模式、运行模式和中断模式。

1. 设计模式 (Design Mode)

启动 VB,即进入设计模式。此时主窗口标题栏显示为“设计”,一个应用程序对应的所有步骤基本上在设计模式下完成,包括程序的界面设计、属性设置和代码编写等,在设计模式下,不能运行程序,也不能使用调试工具但可以设置断点。

2. 运行模式 (Run Mode)

执行“运行”菜单中的“启动”命令(也可单击工具栏的“启动”图标按钮或按 F5 功能键),即进入运行模式。标题栏显示“运行”,在该模式下,不能修改程序代码。若要修改代码,可以选择“运行”菜单的“结束”命令(或单击工具栏中的“结束”图标按钮),回到设计模式进行修改,或者选择“运行”菜单的“中断”命令(也可单击工具栏的“中断”图标按钮或按 Ctrl + Break 组合键),进入中断模式进行修改。

3. 中断模式 (Break Mode)

选择“运行”菜单中的“中断”命令(也可单击工具栏的“中断”按钮或 Ctrl + Break 组合键),程序出现运行错误时,也进入中断模式。标题栏显示“中断”,在此阶段,程序暂停,可以查看并修改代码,检查数据是否正确,修改结束后,可以继续执行程序或中止程序的运行。

在中断模式下,“运行”菜单的“启动”命令变为“继续”命令。

1.3.2 程序错误种类

VB 程序设计中的常见错误可以分为以下四类:语法错误、编译错误、运行错误和逻辑错误。

1. 语法错误 (Syntax Error)

语法错误常常是由于拼错一条命令或使用不正确的语法引起的。例如,关键字拼写不正确,遗漏了某些标点符号,英文的标点符号写成了中文的,函数调用缺少参数,括号不匹配,有 For 没 Next,或把 ElseIf 写成了 Else If 等。语法错误通常在键入程序时发生,是最容易发现的一类错误,因为 VB 具有自动检测语法错误的功能(该功能可以通过“工具”菜单的“选项”命令,在“编辑器”选项卡中设置)。当用户在代码窗口中编辑代码时,VB 会及时检查出语法错误并提示用户纠正。例如以下代码:

```
Private Sub command1_Click()
```

```
a = 9
```

```
b = 4
```

```
c = a * b
```

```
pirnt "c = ";c
```

```
End Sub
```

在输入上述代码时,如果第一行输入为

```
a =
```

按回车键后,就会弹出一错误信息框,同时该语句行变为红色醒目提示,如图 1-8 所示。

此时,若想对程序进行修改,可先单击错误信息框的“确定”按钮(或按回车键)关闭该信息框。如果不明白错误信息的含义,可以单击错误信息框中的“帮助”按钮(或按 F1 键),来获取这条错误产生原因及解决办法的帮助信息。若没有安装 MSDN 将会出现如图 1-9 提示。

使用 VB 的自动语法检测时,需要注意两点: