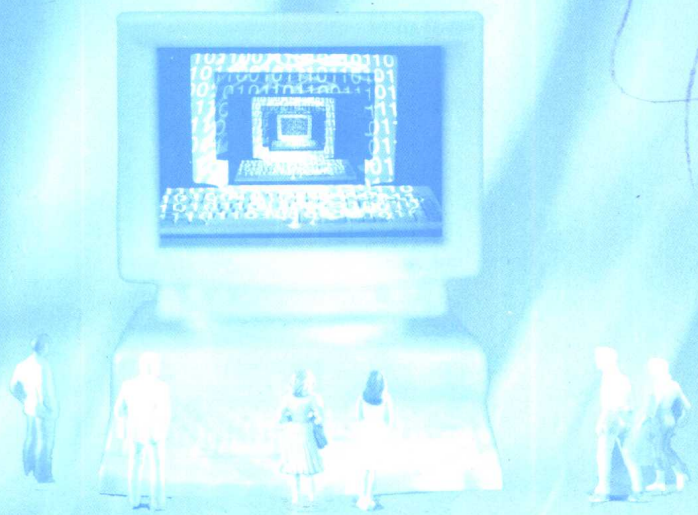
高等教育21世纪课程教材

GAODENG JIAOYU 21 SHIJI KECHENG JIAOCAI

Visual Basic程序设计上机指导与习题选解

VISUAL BASIC CHENGXU SHEJI SHANGJIZHIDAO YU XITIXUANJIE



主编 蒋加伏 张林峰

北京邮电大学出版社
www.buptpress.com

Visual Basic 程序设计上机指导与习题选解

主　编　蒋加伏　张林峰

编　著　朱前飞　黄晓雯　戴小鹏

贺志勇　张红燕　彭小宁

北京邮电大学出版社

内容简介

本书是配合《Visual Basic 程序设计教程》一书编写的 VB 编程实践指导书，全书由三部分组成。第一部分是 Visual Basic 上机指导，第二部分是针对与之配套的教材内容设计的 12 个实验，第三部分为配套教材的习题解答。

本书内容丰富，实践性强，便于自学。

图书在版编目(CIP)数据

Visual Basic 程序设计上机指导与习题选解/蒋加伏，张林峰编著 .—北京：北京邮电大学出版社，2002

ISBN 7—5635—0660—8

Ⅰ.V… Ⅱ①蒋…②张… Ⅲ.BASIC 语言—程序设计—高等学校—教学参考资料 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2002)第 107654 号

书　　名：Visual Basic 程序设计上机指导与习题选解

编　　著：蒋加伏　张林峰

责任编辑：陈露晓

出版发行：北京邮电大学出版社

社　　址：北京市海淀区西土城路 10 号(100876)

电话传真：010—62282185(发行部)　010—62283578(FAX)

E - mail：publish@bupt.edu.cn

经　　销：各地新华书店

印　　刷：国防科技大学印刷厂印刷

开　　本：787mm × 1 092mm　1/16

印　　张：12.25

字　　数：298 千字

版　　次：2003 年 1 月第 1 版　2003 年 1 月第 1 次印刷

ISBN 7—5635—0660—8/TP·92　　定　价：18.00 元

前　言

本书以 Visual Basic 6.0 中文企业版为背景，以 Visual Basic 编程的基础知识为主要内容，从应用的角度出发，重点介绍了 Visual Basic 编程实践中的基本方法和技巧。

全书分为三部分。第一部分是上机指导，这部分主要介绍了 Visual Basic 6.0 中文版联机帮助系统的使用方法，系统、详细地介绍了 Visual Basic 应用程序调试和错误处理的方法及发行应用程序的方法。第二部分为配合教学内容设计了 12 个实验，这部分的每个实验都是针对教材中的重点和难点内容精心设计的。要真正掌握 Visual Basic 程序设计，实践是一个十分重要的环节。第三部分是《Visual Basic 程序设计教程》中习题的参考解答，除能在教材中直接找到答案的部分概念题外，其他类型的习题都给出了参考解答。由于对同一个题目可以编写出多种程序，所以这里给出的答案既不是“标准”答案，更不是“最佳”答案，希望读者在学习的过程中能编写出质量更高的程序。

本书由蒋加伏、张林峰、朱前飞、黄晓雯、戴小鹏、贺志勇、张红燕、彭小宁等编写。

对本书的疏漏和失当，恳请读者批评指正。

编　者

2002 年 12 月

目　录

第一部分　上机指导

第二部分　实验内容

第三部分 习题选解

第一部分　上机指导

1　使用联机帮助系统

Visual Basic 6.0(简称 VB 6.0)为用户提供功能强大的联机帮助系统,它从如何建立一个 VB 程序到如何编辑、编译、调试、运行、发布应用程序都给予了无微不至的帮助。因此,学会使用 VB 的联机帮助系统,就掌握了学习 VB 的一个利器,就可以使用户对在 VB 使用过程中遇到的大量问题迎刃而解。那么,怎样才能有效地使用联机帮助系统呢?

1.1　获取全面的帮助——MSDN Library 查阅器

从 Microsoft Visual Studio 6.0 开始,包括 VB 6.0,所有的帮助文件都采用 MSDN(Microsoft Developer Network)文档的帮助方式。MSDN Library 包含了超过 1GB 的编程技巧信息,其中包括示例代码、开发人员知识库、Visual Studio 文档、SDK 文档、技术文章、会议及技术讲座的论文以及技术规范等,内容非常详细全面。

用户若想查看关于 VB 的全面的帮助信息,需要打开 MSDN Library 查阅器,常见的打开方式有两种:

(1)在 Windows 的"开始"菜单中选择"程序",再在"程序"子菜单下选择"Microsoft Developer Network"子菜单,单击"MSDN Library Visual Studio 6.0(CHS)"。

(2)在 VB 6.0 工作环境中,选择"帮助"菜单的"内容"、"索引"、"搜索"命令之一。

打开后的 MSDN Library 查阅器窗口如图 1－1 所示。

它以浏览器的方式显示帮助文档。窗口分为左、右两个显示区域。在左边区域中,"活动子集"是指用户感兴趣的 MSDN Library 子集,选定子集后,所有的帮助信息只局限于某一内容。例如,要想只获取 VB 的帮助信息,我们可以在下拉式列表框中选择"Visual Basic 文档"。"活动子集"下面有四个选项卡,分别为"目录"、"索引"、"搜索"、"书签",每个选项卡以不同的方式定位和显示帮助文档。

①"目录"选项卡:以目录树结构显示 VB 书的全貌供用户检索和阅读,单击树中的"+"、"－"图标分别打开和关闭其中的帮助主题。

②"索引"选项卡:通过"输入关键字",以索引方式查找帮助信息。

③"搜索"选项卡:通过"输入要查找的单词"在全文中搜索帮助信息。

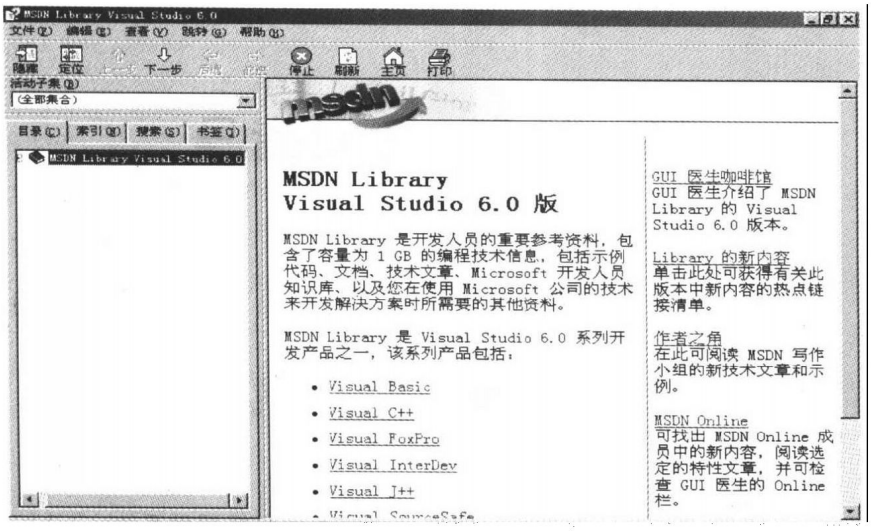


图 1－1 MSDN Library 查阅器窗口

④“书签”选项卡：存放经常使用的某些文档的主题以便快速查找。

以上四个选项卡中，“索引”选项卡通常是使用频率最高的一个。右边区域一般用来显示相应内容。

1.2 获取快捷的帮助——上下文相关帮助

最直接、最快捷的获得帮助信息的方法是使用上下文相关帮助。当用户需要获得窗口、控件、对象、属性、关键词或错误信息的帮助时，只需要选定要帮助的内容，然后按 F1 功能键，VB 便会分析用户在做什么并打开 MSDN Library 查阅器直接定位显示相关的帮助信息。

1.3 获取最新的帮助——Web 上的 Microsoft

MSDN 帮助文档的内容总是在不断更新和补充，要想获得最新、最详细的帮助主题，可以从网上求得帮助。选择“帮助”菜单中的“Web 上的 Microsoft”项，进入网上微软公司的主页，如“常见问题”、“Web 研究会”等，从网上查看或下载最新帮助信息。

1.4 获取深入的帮助——样例应用程序

如果想深入地学习 VB 中某个对象的使用，最好的方法是边看实例边学习。MSDN 提供了上百个 VB 的实例工程，这些实例缺省安装在\Program Files\Microsoft Visual Studio \MSDN98\98VS\2052\Samples\VB98\子目录中。用户可以通过“文件”菜单中的“打开工程”命令，打开所需工程，观察运行结果，分析代码，学习编程思想、方法及技巧。

2　应用程序调试及错误处理

2.1　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。

2.1.1　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”，一个应用程序建立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，只能查看程序代码，不能修改。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)，或者程序出现运行错误时，即进入中断模式。标题栏显示“中断”，在此阶段，程序执行暂停，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。

注意，在中断模式下，“运行”菜单的“启动”命令变为“继续”命令。

2.1.2　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 Form _ Click()
    a=9
    b=4
    c=a * b
    print "c=";c
End Sub
```

在输入上述代码时，如果第一行输入为

```
a=9-
```

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，如图 2－1 所示。

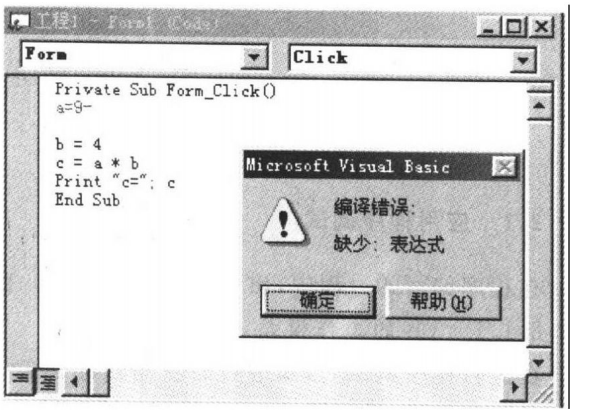


图 2－1 VB 检查语法错误

此时，若想对程序进行修改，必须先单击错误信息框的“确定”按钮(或按回车键)关闭信息框。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。如图 2－2 所示。

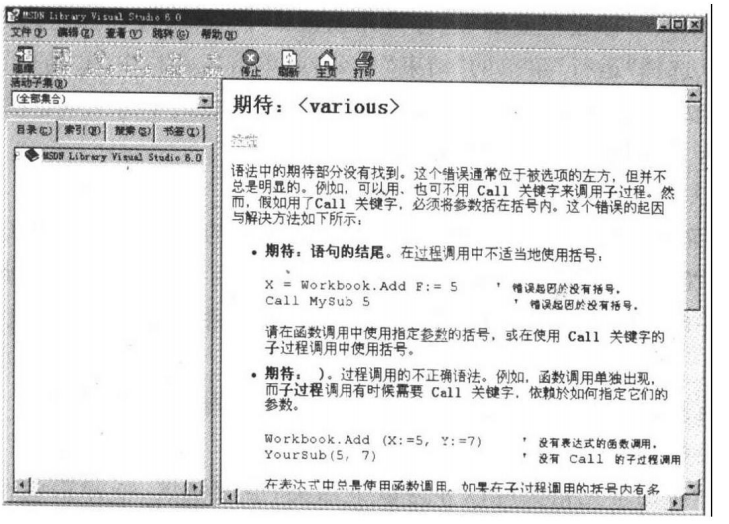


图 2－2 VB 解释错误信息

使用 VB 的自动语法检测时，需要注意两点：

(1)有时错误信息框显示“编译错误”，而非“语法错误”。

(2)一般说来，VB 高亮显示出错单词。但 VB 并不能总是检测到语法错误的确切位置。如果对当前高亮显示的单词看不出问题时，检查前面的一个或两个词，就会发现错误。

2. 编译错误(Compile Error)

编译错误是指选择“运行”菜单的“启动”命令(也可单击工具栏的“启动”图标按钮或按

F5键),或将程序编译成可执行文件时产生的错误。例如,用户未定义变量、遗漏关键字等。这时,VB会弹出一个错误信息框,同时,出错行高亮显示。如图2—3所示。

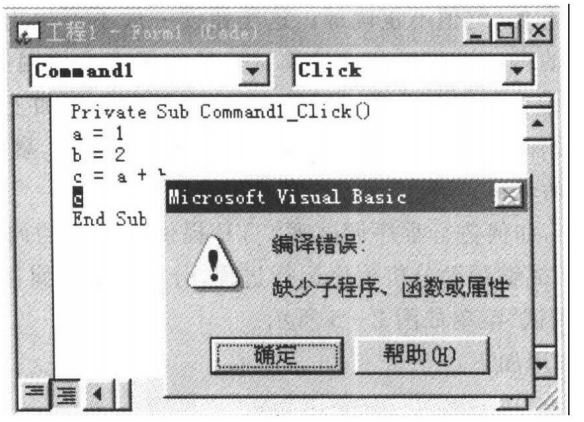


图2—3　编译错误

这种错误不是语法错误,在键入代码时不会被语法检测发现。出现这类错误后,VB将停止编译并回到有错误的代码窗口。

3. 运行错误(Run—Time Error)

运行错误是指程序输入或编译时正确,而在运行代码时发生的错误,这类错误通常是由应用程序在运行期间执行非法操作或某些操作失败引起的。如类型不匹配,用零作除数,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。例如,变量a的类型被定义为单精度型,如果对其赋值的类型是字符串,系统就会发生运行错误,如图2—4所示。

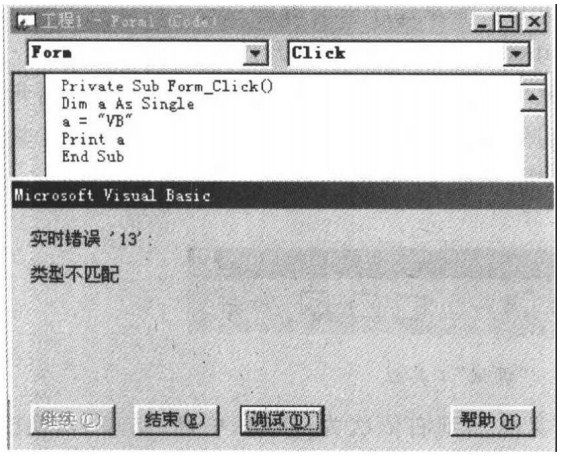


图2—4　运行错误发生

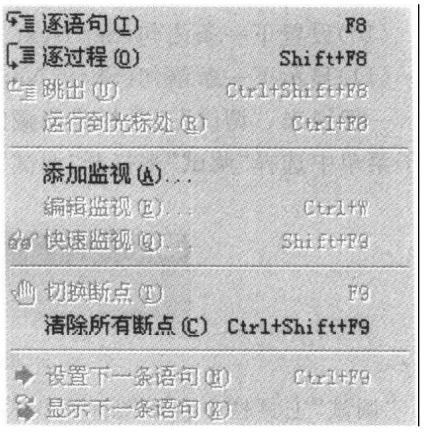


图2—5　"调试"菜单

此时,用户若单击"调试"按钮,则进入中断模式,光标停留在出错行上,可以修改代码;若单击"结束"按钮,则终止程序执行返回到代码窗口;若单击"帮助"按钮,则可以获取有关

错误提示的更多信息。这一类错误在设计阶段较难发现，是程序容错检验的重点。

4. 逻辑错误(Logical Error)

逻辑错误是程序运行后得不到预期结果而产生的错误。例如，运算符使用不正确，语句次序不对，循环语句的起始、终值不正确等。这类错误一般不产生出错信息，因此是最难查找的一类错误，需要认真分析并借助相应的调试工具才能查出原因并加以改正。为减少逻辑错误的发生，良好的编程习惯是非常重要的。如，在关键地方加上必要的注释，强制所有变量必须显式声明从而保证变量名称的一致性，列出详细的事件、事件过程清单等。

2.1.3 使用 VB 的调试工具

在 VB 开发环境中，如何进行程序调试呢？VB 提供了强大的调试工具，即“调试”菜单和“调试”工具栏。它们能够帮助分析程序运行过程，分析变量和属性值的变化，有助于找出程序的错误。VB 的“调试”菜单如图 2—5 所示。

其中各个命令的作用如下：

(1)逐语句(Step Into)：一次执行一条语句；

(2)逐过程(Step Over)：一次执行一个过程或语句；

(3)跳出(Step Out)：执行当前过程的其他部分，并在调用过程的下一行处中断执行；

(4)运行到光标处(Run To Cursor)：通常可以使用这个命令跳过大型循环；

(5)添加监视(Add Watch)：该命令可以弹出“添加监视”对话框，可以在该对话框中输入出现在“监视”窗口的监视表达式；

(6)编辑监视(Edit Watch)：该命令弹出“编辑监视”对话框，通过此对话框可以编辑或删除监视表达式；

(7)快速监视(Quick Watch)：将所选择的表达式值显示在“快速监视”对话框内；

(8)切换断点(Toggle Breakpoint)：用于设置或删除当前行上的一个断点；

(9)消除所有断点(Clear All Breakpoints)：消除程序中所有断点；

(10)设置下一条语句(Set Next Statement)：选定语句作为开始执行行；

(11)显示下一条语句(Show Next Statement)：高亮显示下一条将要被执行的语句。

一般说来，“调试”工具栏是隐藏的，选择“视图”菜单中的“工具栏”菜单项，在工具栏下的子菜单中选择“调试”来打开“调试”工具栏，如图 2—6 所示。

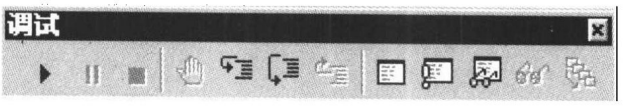


图 2—6 “调试”工具栏

“调试”工具栏中各个图标所代表的含义从左到右依次为：启动、中断、结束、切换断点、逐语句、逐过程、跳出、本地窗口、立即窗口、监视窗口、快速监视和调用堆栈。其中启动、中断、结束三个按钮控制程序的运行、中断和结束；切换断点、逐语句、逐过程、跳出与调试菜单中相应命令作用相同；本地窗口用于观察当前过程中所有变量的值；立即窗口用于中断状态下显示和改变变量、表达式或对象属性的值；快速监视用于快速查看、选中变量、表达式或对

象属性的值；调用堆栈用于显示当前所有活动过程调用的一个列表，该列表显示的项目是以分级格式显示的，可以跟踪程序的进程。

1. 进入中断模式

调试工具只有在应用程序的中断模式下才有效。因此要使用调试工具，首先应该进入中断模式，此模式的作用是暂停应用程序的执行并提供有关应用程序的情况，程序总是从运行模式进入中断模式。

进行程序调试时，常用的中断方法有两种：设置断点和使用 Stop 语句。

(1)设置断点。

进入中断模式最准确和最常用的方法是设置断点。通常在需要程序暂停的地方设置断点，断点可以设置多个，用来缩小错误查找范围，对程序分段进行调试。

断点一般在中断模式或设计模式下设置。设置断点的方法如下：

①在代码窗口中希望中断的语句行左边的灰色区域(称边界标识条)，单击鼠标，如图 2—7 所示。此时，该语句行以粗体反相显示。

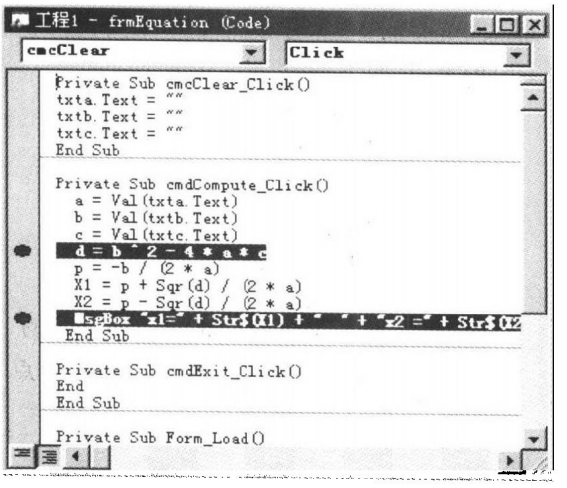


图 2—7　设置断点

②在代码窗口中，将光标移到希望中断的语句上，选择“调试”菜单中的“切换断点”命令，单击“调试”工具栏的“切换”按钮或按 F9 快捷键。

上述方法实际上是切换断点。即无断点时设置断点，有断点时清除断点。需要注意的是，清除断点时，只需重复设置断点时的操作即可。若设置了多个断点，可以选择“调试”菜单中的“清除所有断点”命令或按 Ctrl＋Shift＋F9 组合键。设置断点后，运行程序，当程序执行到断点处，暂停程序进入中断模式，黄色高亮显示断点语句。此时可以查看断点语句(该语句未执行)之前的所有变量、属性、表达式的值，方法是将鼠标指向所关心的变量处，稍停一下，鼠标下方就会显示该变量的值，如图 2—8 所示。

(2)使用 Stop 语句。

设置断点的另一种方法是在程序代码中需要暂停的地方加上一个或多个 Stop 语句。当程序遇到 Stop 语句时，就会暂时停止该程序的执行，并进入中断模式以便调试。例如，有

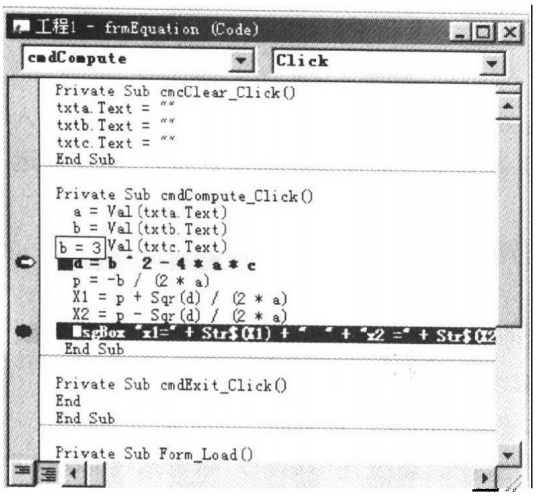


图 2－8 查看变量 b 的值

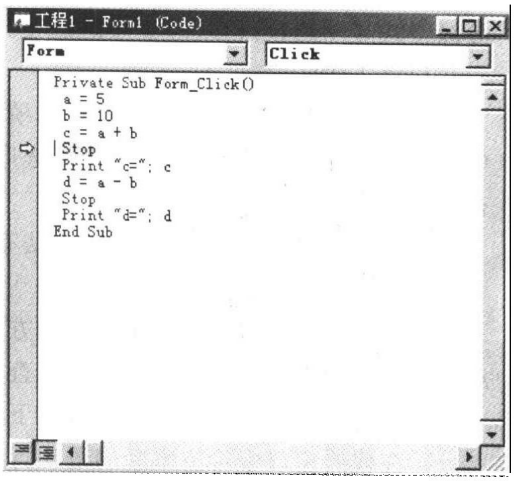


图 2－9 使用 Stop 语句

程序如下：

```
Private Sub Form _ Click()
    a=5
    b=10
    c=a+b
    Stop
    Print "c=";c
    d=a-b
    Stop
    Print "d=";d
End Sub
```

运行该程序，结果如图 2－9 所示。

需要注意的是，Stop 语句和 End 语句是不同的。Stop 语句只是暂停程序的运行，只要再选择“继续”命令程序就可继续执行，而 End 语句则是终止整个程序的执行。

(3)设置断点和使用 Stop 语句的比较。

两者的相同点是：作用是一样的，即暂停程序执行并进入中断模式。中断后，都可通过“运行”菜单中的“继续”或按 F5 键继续执行。

两者的不同点是：断点较 Stop 方便，因为无须修改代码，但 Stop 比断点灵活，因为它可以使程序在一定条件下暂停。如程序段：

```
If a+b<c Or a+c<b Or b+c<a Then
    Stop
End If
```

2. 程序跟踪

利用断点，只能查出错误大致发生的程序段，要确定是哪一条语句发生问题，需要利用程序跟踪。VB 调试工具提供四种跟踪方式：逐语句执行、逐过程执行、跳跃执行和运行到光标处。

(1)逐语句执行。

所谓逐语句执行，就是每次只执行一条语句，根据输出结果来判断执行语句是否正确。也称为单步执行。将设置断点和逐语句跟踪相结合，是初学者调试程序最简捷的方式。要进行逐语句执行，可以选择调试菜单中的“逐语句”命令(也可单击调试工具栏中的“逐语句”按钮或按 F8 键)。

启动逐语句执行命令后，程序进入运行模式。每执行完一条语句后，切换到中断模式并高亮显示下一条语句。如图 2－10 所示。因此可以知道程序的执行顺序及当前的执行位置，从而查找出错误语句。

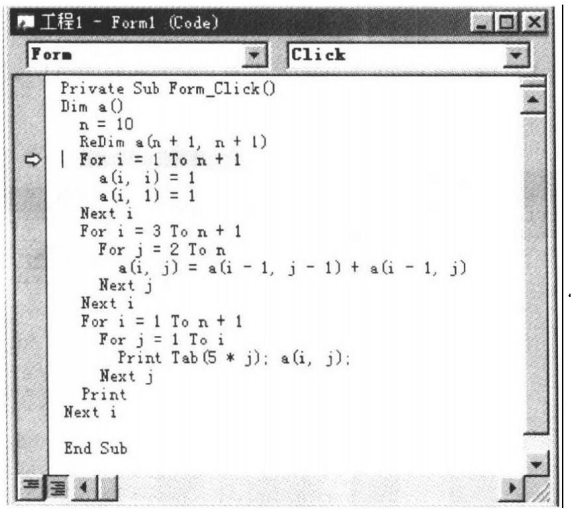


图 2－10　逐语句执行

需要注意的是：逐语句执行以语句为单位，不是以行为单位，因此当一行中有多条语句时，每次只有一条语句高亮显示。另外，如果执行的是事件过程，一定要触发事件才能执行事件过程中的代码。例如，执行 Form _ Click 事件过程，单步执行后，屏幕上显示窗体，此时必须单击窗体，才能开始执行。其他事件过程，与此类似。

(2)逐过程执行。

逐过程执行指一次执行一个过程或语句。它将“过程”等同为语句，一次执行完毕。当面对一个已经调试过的过程，没有必要再一句句执行时，逐过程执行是很有用的。

要进行“逐过程”执行时，可以选择“调试”菜单中的“逐过程”命令(也可以单击"调试"工具栏中的“逐过程”按钮或按 Shift＋F8 键)

例如，有两个通用过程：

```
Sub question()
```

```
    Print "VB is fun?"
End Sub
Sub answer()
    Print "VB is fun!"
End Sub
```

在 Form_Click 事件过程中调用上述过程：

```
Private Sub Form_Click()
    Call question
    Call answer
End Sub
```

启动“逐过程”执行，屏幕显示窗体，单击窗体后，执行窗体单击事件过程，Private Sub Form_Click()高亮显示，再次启动“逐过程”，Call question()高亮显示，准备执行，如图 2－11 所示，再按一次 Shift＋F8 键，窗体上显示“VB is fun?”，同时，Call answer()高亮显示，准备执行，继续按 Shift＋F8 键，过程执行，窗体上显示“VB is fun!”，同时“End Sub”高亮显示，程序执行结束。

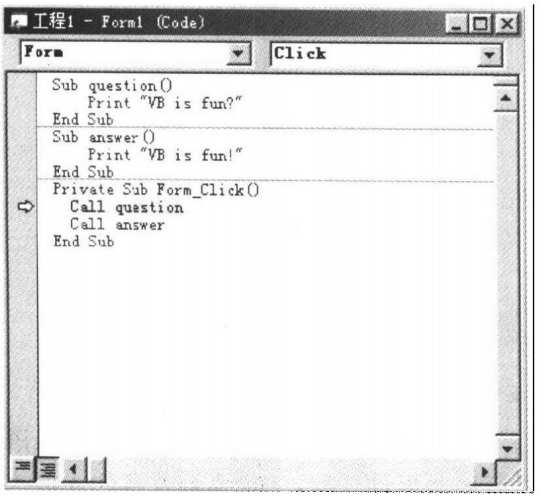


图 2－11 逐过程执行

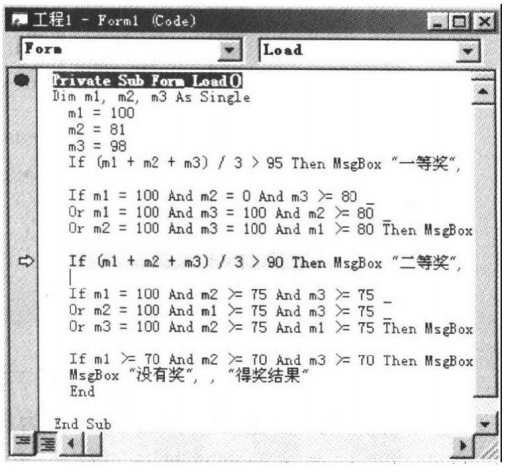


图 2－12 跳跃执行

(3)跳跃执行。

逐语句执行和逐过程执行都只能按顺序一次执行一条语句或一个过程。如果想改变这种方式，例如，暂时避开程序的某一部分，调试其他部分，或者对程序修改后再回过头来执行，则必须通过跳跃执行来实现。

要进行跳跃执行，可以首先启动程序，进入中断模式，此时边界标识条中有一箭头指向下一个被执行语句。然后另外选择一条要执行语句，如“If (m1 + m2 + m3) / 3 > 90 Then MsgBox “二等奖”, , “得奖结果”: End”。其次，选择“调试”菜单中的“设置下一条语句”命令(或按 Ctrl＋F9 键)，把该行设置为开始执行行，如图 2－12 所示。最后，继续运行

程序(用“运行”中的“继续”命令或按 F5 键),则在原执行行和新执行行之间的语句段被忽略。注意,“设置下一条语句”命令只能在当前过程中使用。

(4)运行到光标处。

在设计阶段,在代码窗口中将光标移到某一行上,然后执行“调试”菜单中的“运行到光标处”命令(或按 Ctrl+F8 键),程序将会在光标所在行停止运行,并在边界标识条中显示相应的标记,如图 2-13 所示。

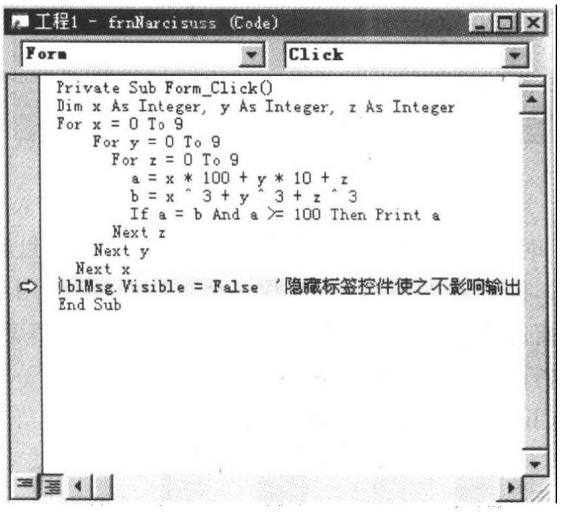


图 2-13　运行到光标处

用“运行到光标处”命令可以跳过大型循环。

注意,光标所在行必须在程序的执行流程中。

3. 调试窗口

在中断模式下,除了用鼠标指向要观察的变量、属性和表达式直接显示其值外,还可以通过调试窗口来观察有关变量、属性和表达式的值。VB 主要有三种窗口提供调试,即“监视窗口”、“立即窗口”和“本地窗口”。可单击“视图”菜单中的相应命令或“调试”工具栏中的相应按钮来打开这些窗口。

(1)“监视”窗口。

在“监视”窗口中只能被动地显示变量或表达式的值。在此之前,必须在设计模式下利用“调试”菜单中的“添加监视命令”或“快速监视”命令添加监视表达式以及设置监视类型,则运行时“监视窗口”根据所设置的监视类型进行相应显示。选择“调试”菜单中的“添加监视”命令后,屏幕上弹出一个对话框,如图 2-14 所示。

该对话框分为三部分:“表达式文本框”、“上下文”和“监视类型”。文本框用来输入监视表达式,表达式可以是算术表达式、关系表达式、逻辑表达式等。“上下文”部分指定要监视的过程和模块。“监视类型”包括三个单选按钮,可根据需要进行选择来设置监视类型。依据监视类型,可将监视分为监视点和监视表达式。当选择类型为“当监视值为真时中断”