



普通高等教育农业部“十二五”规划教材
全国高等农林院校“十二五”规划教材

Visual Basic

程序设计案例实验教程

张林峰◎主编



中国农业出版社

普通高等教育农业部“十二五”规划教材
全国高等农林院校“十二五”规划教材

Visual Basic 程序设计

案例实验教程

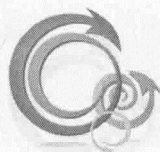
张林峰 主编

中国农业出版社

内容简介

本书是与《Visual Basic 程序设计案例教程》相配套的实验教材。全书分三部分，第一部分为上机指导，主要介绍 Visual Basic 帮助系统的获取、程序的调试和错误处理、应用程序的发布；第二部分为案例实训，针对配套教材的内容和进度安排了 11 类精选实验和 1 个综合实验；第三部分为综合练习。

本书可作为高等学校非计算机专业 Visual Basic 程序设计课程的实验教材，也可用作计算机水平考试和社会读者的自学辅助用书。



编写人员名单

主 编 张林峰

副主编 朱幸辉 谭湘键 向昌盛

参 编 (以姓名笔画为序)

龙陈锋 吴 伶 拜战胜

熊大红 张红燕 贺细平

全书共分上机指导、案例实训和综合练习三部分。

1. 上机指导 主要介绍 Visual Basic 联机帮助系统的获取、程序调试方法和技巧, 如何及时发现并排除程序中的错误, 以及如何发布应用程序。这一部分对读者开发 Visual Basic 应用程序非常有用, 读者应充分重视。

2. 案例实训 本部分是本书的核心内容, 每个案例都针对某个具体应用精心设计, 案例内容力求突出典型性、实用性和趣味性, 部分实训做了详细的分析, 给出了设计过程、源程序, 部分实训则只给出要求, 希望读者自己动手、动脑, 独立完成, 以达到举一反三的学习效果。实训内容均选自教学大纲, 既有基础型实训题, 也有综合型实训题。

3. 综合练习 主要是为了巩固读者对前面所学的基础知识, 强化编程能力。

由于时间仓促, 加之作者水平有限, 书中疏漏和不足之处恳请读者批评指正, 以便不断完善。



前 言

《Visual Basic 程序设计案例实验教程》是《Visual Basic 程序设计案例教程》教材的配套用书，是作者在十余年的实践教学基础上经过精心挑选、反复提炼形成的。

本书精选的所有案例遵循由浅入深、循序渐进的原则，突出趣味性、应用性和可操作性，将知识点融会于各个案例中，使读者在学习过程中不但能掌握 Visual Basic 程序设计的较完整的知识体系，还能提高分析问题和解决问题的能力，同时能激发读者的求知欲，提高学习的兴趣。

全书共分为上机指导、案例实训和综合练习三部分：

1. 上机指导 主要介绍 Visual Basic 联机帮助系统的获取、程序调试方法和技巧，如何及时发现并排除程序中的错误，以及如何发布应用程序。这一部分内容对读者开发 Visual Basic 应用程序非常有用，读者应充分重视。

2. 案例实训 本部分是本书的核心内容，每个案例都针对某个具体应用精心设计。实验内容力求突出典型性、实用性和趣味性，部分实验做了详细的分析，给出了设计过程和源代码，部分实验则只给出要求，希望读者自己动手、动脑，独立完成，以达到举一反三的学习效果。实验内容与理论教学同步，能有效地配合理论教学，使学生通过实验能够融会贯通并且灵活应用理论知识。

3. 综合练习 主要是为了让读者复习巩固所学的基础知识，强化编程能力。

由于时间仓促，加之作者水平有限，书中疏漏和不当之处恳请读者批评指正，以便不断完善。

编 者

2011 年 11 月



目 录

前言

第一部分 上机指导

第 1 章 获取帮助	1
1.1 通过 MSDN Library 获得帮助	1
1.2 使用上下文相关帮助	2
1.3 通过 Internet 获得帮助	2
1.4 使用样例应用程序获得帮助	3
第 2 章 程序调试和错误处理	4
2.1 Visual Basic 工作模式	4
2.1.1 设计模式	4
2.1.2 运行模式	4
2.1.3 中断模式	5
2.2 Visual Basic 错误类型	5
2.2.1 语法错误	5
2.2.2 编译错误	6
2.2.3 运行错误	6
2.2.4 逻辑错误	6
2.3 Visual Basic 程序调试	6
2.3.1 “调试”工具栏	7
2.3.2 标识当前模式	7
2.3.3 使用调试窗口	8
2.3.4 使用中断	11
2.3.5 程序跟踪	12
2.4 Visual Basic 错误处理	14
2.4.1 错误的捕捉及错误处理子程序	14
2.4.2 Error 函数与 Error 语句	15
2.4.3 设计错误处理程序	16
第 3 章 程序发布	18

第二部分 案例实训

实验一 简单 Visual Basic 程序设计	21
【实验 1-1】熟悉 Visual Basic 6.0 集成开发环境	21

【实验 1-2】电子台历 1	23
【实验 1-3】电子台历 2	24
【实验 1-4】动态设置文字效果	24
【实验 1-5】问候程序 1	25
【实验 1-6】问候程序 2	26
自己动手	27
(1) 电子台历 3	27
(2) 问候程序 3	27
实验二 顺序结构程序设计	28
【实验 2-1】求圆柱体积	28
【实验 2-2】求三角形面积	29
【实验 2-3】美元/人民币金价换算器	30
【实验 2-4】工资钞票数量计算器	31
【实验 2-5】一元二次方程求解 1	32
自己动手	34
(1) 一元二次方程求解改进	34
(2) 问候程序 4	34
实验三 选择结构程序设计	35
【实验 3-1】个人收入所得税计算器	35
【实验 3-2】问候程序 5	36
【实验 3-3】问候程序 6	37
【实验 3-4】一元二次方程求解 2	38
【实验 3-5】三个数升序排列	39
自己动手	40
(1) 三个数降序排列	40
(2) 闰年的判定	40
实验四 循环结构程序设计	41
【实验 4-1】累加	41
【实验 4-2】输出可见标准 ASCII 码表	43
【实验 4-3】矩阵运算	44
【实验 4-4】摇奖程序	45
【实验 4-5】成绩统计	47
自己动手	49
(1) 条件累加	49
(2) 累加累乘	49
(3) 九九乘法表	50
实验五 常用算法程序设计 (一)	51
【实验 5-1】4 位反序数	51
【实验 5-2】抓交通肇事车	52
【实验 5-3】阅卷问题	53

【实验 5-4】验证哥德巴赫猜想	53
【实验 5-5】分数段人数统计	54
自己动手	55
(1) 五角星数	55
(2) 求零件数	55
实验六 常用算法程序设计 (二)	56
【实验 6-1】杨辉三角形	56
【实验 6-2】冒泡法排序	57
【实验 6-3】统计字符 (串) 出现的次数	58
【实验 6-4】合并两个有序数组	59
【实验 6-5】递归三角形	60
自己动手	61
(1) 降序排序	61
(2) 求满足条件的素数个数	61
(3) 求满足条件的四位正整数	61
实验七 可视化程序设计 (一)	62
【实验 7-1】密码校验	62
【实验 7-2】简单计算器	63
【实验 7-3】简单记事本程序	64
【实验 7-4】简单选课系统	68
自己动手	70
(1) 密码校验程序改进	70
(2) 为简单计算器添加选单	70
实验八 可视化程序设计 (二)	71
【实验 8-1】简单学生信息输入系统	71
【实验 8-2】会议记录	73
【实验 8-3】收款程序	76
自己动手	78
(1) 歌词同步 Show 的歌词制作程序	78
(2) Flash 播放器改进	78
实验九 图形程序设计	79
【实验 9-1】万花筒	79
【实验 9-2】分色程序	80
【实验 9-3】简单动画: 红绿灯	81
【实验 9-4】简单动画: 碰撞球	82
【实验 9-5】简单绘图程序	83
自己动手	84
(1) 放射彩线	84
(2) 渐变色	84
实验十 数据库程序设计 (一)	85

【实验 10-1】用网格控件浏览数据库记录	85
【实验 10-2】条件查询	86
【实验 10-3】组合查询	88
【实验 10-4】组合查询与排序	91
自己动手	94
(1) 修改查询条件	94
(2) 电子相簿	94
实验十一 数据库程序设计 (二)	96
【实验 11-1】模拟词霸词库生成程序	96
【实验 11-2】学生信息浏览、输入程序 1	98
【实验 11-3】学生信息浏览、输入程序 2	99
自己动手: 学生信息浏览、输入程序 3	102
实验十二 综合应用——学生成绩管理系统	103
一、系统设计	103
二、数据库设计	103
三、系统实现	104
自己动手: 实用无纸化考试系统	111

第三部分 综合练习

综合练习一	112
综合练习二	118
综合练习三	123
综合练习一参考答案	130
综合练习二参考答案	131
综合练习三参考答案	133

主要参考文献	135
--------------	-----

第一部分 上机指导

第 1 章 获取帮助

在用程序设计语言开发应用程序时，经常会遇到各种问题，如何有效获取帮助是非常重要的。本章主要介绍在 Visual Basic 中如何有效获取帮助。Visual Basic 系统为用户提供了较为完备的帮助系统，Visual Basic 6.0 的帮助系统采用了 MSDN (Microsoft developer network) Library 的帮助方式，MSDN Library 涵盖了微软全套可开发产品线的技术开发文档和科技文献（部分包括源代码），也包括过刊的 MSDN 杂志节选和部分经典书籍的节选章节。用户在安装时可以选择安装 MSDN 帮助文件，用户通过 MSDN 可以获得非常实用的帮助信息，特别是其中的示例程序，对初学者提高学习 Visual Basic 的效率、解决编程中出现的问题有很大的裨益。

1.1 通过 MSDN Library 获得帮助

(1) MSDN Library 启动：

- 启动 Visual Basic。
- 在 Visual Basic 窗口中选择“帮助”选单，在下拉选单中选择“内容”、“索引”或“搜索”命令，都可打开“MSDN Library Visual Studio 6.0”窗口，如图 1-1-1 所示。
- MSDN 以浏览器的方式显示帮助文档，它保持了浏览器的全部特性。窗口下半部分的左侧有 4 个选项卡打开不同的选项卡可以获得不同方式的帮助。

(2) MSDN Library 帮助方式。

- 目录方式：打开“目录”选项卡，可以在左显示区列表中显示所有文档的名称。单击某个目录，则在窗口的右侧显示相应的内容。
- 索引方式：打开“索引”选项卡，则可在“键入要查找的关键字”文本框中输入要查找的内容，在其下面的列表框中显示查找到的关键字，单击该关键字，然后单击“显示”按钮，即可在窗口右侧显示相应的内容。
- 搜索方式：打开“搜索”选项卡，在“键入要查找的单词”文本框中输入要查找的单词，单击“列出主题”按钮，即可在下面的“选择主题”列表中列出相关的主题，选择其中的某个主题，单击“显示”按钮，将在窗口右侧显示区显示该主题的内容。
- 书签方式：打开“书签”选项卡，在“主题”列表中选择某个主题，然后单击“显示”按钮，即可在窗口右侧显示该主题的内容。



图 1-1-1 MSDN 帮助窗口

1.2 使用上下文相关帮助

Visual Basic 允许使用上下文相关帮助，即不用通过“帮助”选单就可直接获得有关这些内容的帮助信息，大大方便了用户的使用。例如，要获得有关 Visual Basic 语言中任何关键词的帮助信息，则只需将光标插入点置于代码窗口中的关键词的任意位置按 F1 键，即可打开帮助窗口，在窗口右侧显示该关键词的帮助信息。

Visual Basic 中的以下对象均可使用上下文相关帮助：

- Visual Basic 中的每个窗口，如“属性”窗口、代码窗口、“工程资源管理器”窗口等。
- 工具箱中的控件。
- 窗体或文档对象中的对象。
- “属性”窗口中的属性。
- Visual Basic 关键词，包括过程、声明、函数、属性、方法、事件和特殊对象的名字。
- 错误信息。

1.3 通过 Internet 获得帮助

MSDN Library 中有许多内容直接链接至外部网站，以方便获得更多帮助。除此之外，Internet 上有许多用于学习和交流的 Visual Basic 站点，通过这些站点可以与世界各地的 Visual Basic 爱好者互相学习和交流，获取软件开发方面的资料 [如微软公司的 Visual Basic 官方网站 (<http://msdn.microsoft.com/vbasic/>)、Visual Basic 爱好者乐园 (<http://www.vbgood.com/>)等]]。

1.4 使用样例应用程序获得帮助

Visual Basic 软件包中提供了大量的实例，为学习、理解、掌握 Visual Basic 提供了很大的方便。除在帮助系统中提供了大量示例程序代码（图 1-1-2）外，在 MSDN 中还提供了许多实例工程，这些实例默认安装在“\Program Files\Microsoft Visual Studio\MSDN 98\98 VS\2052\Samples\Visual Basic 98\”子目录中，用户只要打开所需的工程就可以运行并观察其效果，也可查看代码，学习各控件的使用和编程方法。



图 1-1-2 MSDN “帮助”中的代码示例

第 2 章 程序调试和错误处理

Visual Basic 是创建 Windows 应用程序最为简捷、便利的工具之一，是一种可视化、面向对象、采用事件驱动方式的结构化高级程序设计语言。Visual Basic 虽然简单，但在实际编程过程中难免会出现错误，出了错误就必须改正。通过不断改正，让程序能正常运行并得到预期结果的过程就是程序调试。本章介绍 Visual Basic 中程序调试的相关内容。

2.1 Visual Basic 工作模式

Visual Basic 有三种不同的工作模式，即设计模式、运行模式和中断模式，它们是编写、调试和运行应用程序时应用程序所处的不同状态。

2.1.1 设计模式

启动 Visual Basic 新建工程或打开一个工程后，打开一个“工程”窗口，即进入设计模式，在主窗口标题栏上显示“[设计]”字样，如图 1-2-1 所示。

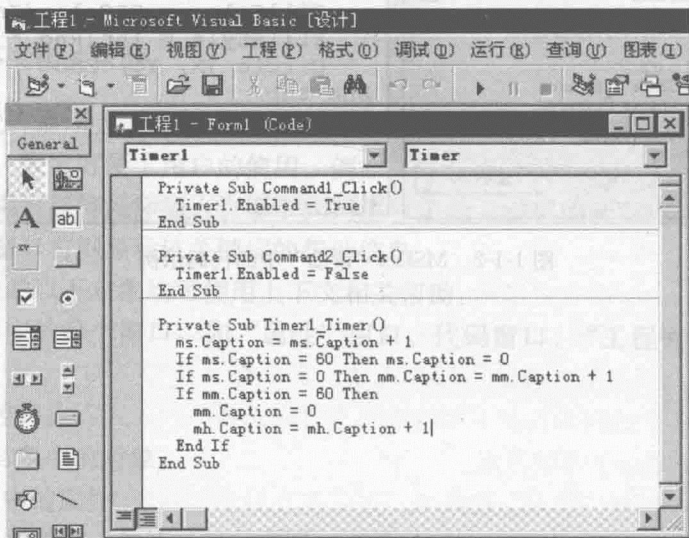


图 1-2-1 设计模式窗口

创建 Visual Basic 应用程序的所有工作都是在设计模式下完成的。在设计模式状态下，用户可以设计窗体，绘制控件，编写程序代码，并使用“属性”窗口设置属性值或查看当前属性值等。当程序处于设计模式时，不能执行操作，也不能使用调试工具对之进行调试，但可设置断点和创建监视表达式。

2.1.2 运行模式

在一个 Visual Basic 工程中，选择“运行” | “启动”命令（或按 F5 键，或单击工具栏中的“运行”按钮），应用程序即可进入运行模式，此时主窗口标题栏上原来显示的“[设

计]”被“[运行]”取代。

在执行模式中，Visual Basic 将全部控制权交给应用程序，用户可以对应用程序进行测试，可查看程序代码，检验程序的运行结果，但不能修改代码。

选择“运行”|“结束”命令（或单击工具栏中的“结束”按钮），可返回设计模式。选择“运行”|“中断”命令（或单击工具栏中的“中断”按钮，或按 Ctrl+Break 组合键），可进入中断模式。

2.1.3 中断模式

当程序处于中断模式时（暂停程序的执行），即进入 Visual Basic 中断模式，此时主窗口标题栏中原来显示的“[设计]”或“[运行]”由“[Break]”取代。在中断模式下，因为变量和属性设置值被保留下来，所以，可以分析应用程序的当前状态，并可修改变量或属性的值，这些修改将影响程序的运行。此时，可以在应用程序中修改代码，观察应用程序界面的情况，确定哪个过程已被调用，监视变量和属性值的变化，查看或控制应用程序下一步运行的语句，也可在“立即”窗口中执行 Visual Basic 语句，手工控制应用程序的操作。

Visual Basic 中可通过以下 4 种方式进入中断模式：

- 在运行模式下，选择“运行”|“中断”命令。
- 在程序中设置断点（break point），程序执行到该断点时自动进入中断模式。
- 执行程序时遇到了 Stop 语句。
- 在程序执行过程中，如果出现错误，则自动进入中断模式。

2.2 Visual Basic 错误类型

任何用户在编写程序时，都难免出现这样或那样的错误，这些错误称为 Bug。找出并纠正这些错误的过程称为 Debug（程序调试）。

Visual Basic 应用程序的错误一般有语法错误、编译错误、运行错误和逻辑错误四类。

2.2.1 语法错误

Visual Basic 语法错误是由于创建了不正确的代码，即在语法不正确时出现这种错误。如错误地输入了关键字，丢失或写错了符号，遗漏了必需的语句成分，括号不匹配等。

Visual Basic 具有自动语法查错功能，在设计阶段输入程序代码时就能检查出语法错误。例如以下代码：

```
Private Sub Command1_Click()  
    Num1 = 10; Num2 = 20  
    Sum = Num1 + Num2  
    Print Sum  
End Sub
```

在输入代码时，若将第 3 行输入

```
Sum = Num1(Num2
```

按 Enter 键后就会显示出错提示框，并且刚输入的一行变成高亮显示，单击“确定”按钮后，错误行代码变成红色，如图 1-2-2 所示。



图 1-2-2 语法错误提示

只有改正该语法错误后，红色代码才会消失。

2.2.2 编译错误

在运行 Visual Basic 程序前，系统将自动对源程序进行编译，在编译过程中出现的错误称为编译错误，Visual Basic 对编译中产生的错误将进行提示。如果用户未定义变量或遗漏关键字等，都将给出错误提示，并使有错误的程序行高亮显示，如图 1-2-3 所示。出现这类错误后，Visual Basic 将停止编译，并返回有错误的程序代码窗口。

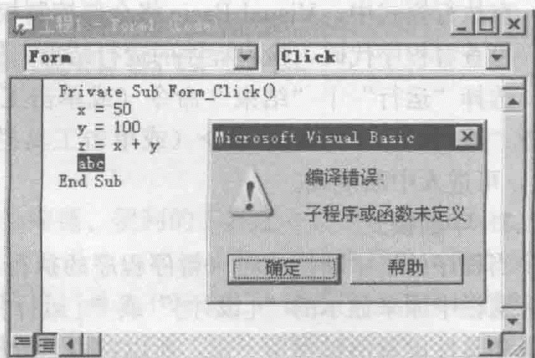


图 1-2-3 编译错误提示

2.2.3 运行错误

虽然语法正确，但程序运行期间，如果一个语句企图执行一个不能执行的操作，就会产生运行错误。有运行错误的代码在一般情况下运行正常，但遇到非法数据或系统条件禁止代码运行时就会产生错误。一个典型的运行错误是以 0 作为除数，运行时出现的错误提示如图 1-2-4 所示。

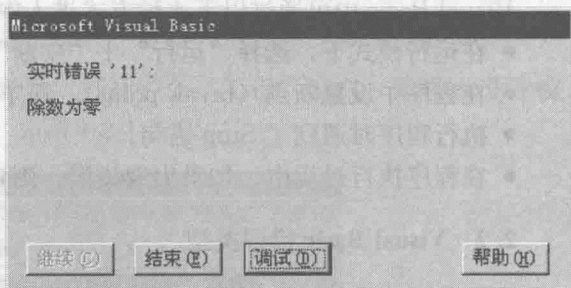


图 1-2-4 运行错误提示

2.2.4 逻辑错误

程序的逻辑错误是指应用程序未按设计逻辑进行运行，或生成了无效甚至错误的结果。这种错误是由于程序代码中不恰当的逻辑设计而引起的，与语法错误和运行错误不同，逻辑错误一般不报告出错信息。它既没有语法错误，也没有运行错误，从表面上看一切正常，但会得到错误的结果，例如，用户将 1.05 误写为 10.5，这种错误只有对运算结果进行分析才能发现和排除。这类错误最难发现，因而也最危险。要减少或克服逻辑错误，没有捷径可寻，只能靠耐心、经验以及良好的编程习惯。

(1) 编程时减少逻辑错误的常用办法。

- 列出与应用程序有关的所有事件，然后考虑如何在程序中响应这些事件，认真定义每个事件过程及通用过程。
- 在关键地方加上必要的注释。
- 注意变量名的一致性。

(2) 排除逻辑错误的一般方法。

- 设置断点，缩小错误的查找范围。
- 中断程序的运行，检查变量的当前值。

值得注意的是，在设计过程中，一般来说，计算机不会出错，出错的往往是人。

2.3 Visual Basic 程序调试

Visual Basic 为程序调试提供了一组交互的、有效的调试工具，这些调试支持包括设置

断点，逐语句、逐过程地控制程序的运行，利用调试窗口显示监视表达式、变量的值和属性的值等。使用这些调试工具，可以快捷、有效地检查程序中非语法或语义错误产生的地点和原因，并加以纠正。

2.3.1 “调试”工具栏

Visual Basic 调试工具包括断点、中断表达式、监视表达式、一次执行一条语句或一个过程、显示变量的值或属性的值等。在各种调试工具中，“调试”工具栏提供了绝大多数调试工具的访问。如果在屏幕中没有“调试”工具栏，则可选择“视图”|“工具栏”|“调试”命令，或在 Visual Basic 工具栏中右击，在弹出的快捷选单中选择“调试”命令，打开“调试”工具栏，如图 1-2-5 所示。



图 1-2-5 “调试”工具栏

工具栏中的按钮从左到右依次为“启动”按钮、“中断”按钮、“结束”按钮、“切换断点”按钮、“逐语句”按钮、“逐过程”按钮、“跳出”按钮、“本地窗口”按钮、“立即窗口”按钮、“监视窗口”按钮、“快速监视”按钮和“调用堆栈”按钮。对应的所有调试工具的功能见表 1-2-1。

表 1-2-1 调试工具的功能

工 具	功 能
启动	启动应用程序
中断	中断程序
结束	结束应用程序的运行
切换断点	在光标所在行设置断点
逐语句	单步执行
逐过程	单步执行可执行语句，但不单步执行调用过程中的语句
跳出	执行该过程的剩余代码，在下一个过程第一行中断
本地窗口	显示本地变量的值
立即窗口	在程序中中断方式下，可以执行代码或查询值
监视窗口	显示选中的表达式的值
快速监视	在程序中中断方式下，列出表达式的当前值
调用堆栈	在中断方式下，显示所有被调用而未返回的过程

说明：应当注意，“逐过程”和“逐语句”调试工具有所不同，“逐语句”工具单步执行所有可执行代码，也包括在过程中调用的其他过程，而“逐过程”则不单步执行该过程中调用的其他过程。

2.3.2 标识当前模式

在 Visual Basic 的标题栏中总是显示工程的当前模式，可以很容易识别出当前所处模式，如图 1-2-6、图 1-2-7 和图 1-2-8 所示。

可以通过选择“运行”选单中的命令或单击工具栏中的命令按钮在 3 种模式间切换。例

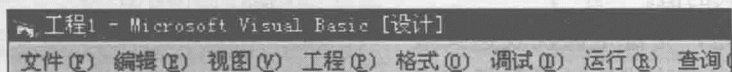


图 1-2-6 设计模式标题栏

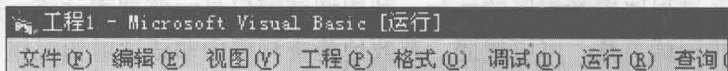


图 1-2-7 运行模式标题栏



图 1-2-8 中断模式标题栏

如，处于设计模式时，在“运行”选单中选择“启动”命令，或单击“运行”按钮，可切换到运行模式。选择“运行”|“结束”命令，或单击“结束”按钮，则返回设计模式。应用程序处于运行模式时，选择“运行”|“中断”命令，或单击“中断”按钮，或按 Ctrl+Break 组合键，可切换到中断模式。此外对于发生以下情况时，将自动切换到中断模式：

- 语句产生了非俘获的运行错误。
- 语句产生运行错误，并且“发生错误时中断”错误捕获选项被选中。
- “添加监视”对话框中定义的中断表达式将发生变化或变成真，这取决于它的定义方式。
- 执行到一个设有断点的行。
- 执行到一个 Stop 语句。

2.3.3 使用调试窗口

在中断模式下，用户可以通过“监视”窗口、“立即”窗口和“本地”窗口等这些调试窗口来观察相关变量的值。选择“视图”菜单中的相应命令可以打开上述调试窗口。

1. “监视”窗口 “监视”窗口用于显示当前的监视表达式，用监视表达式可以查看或跟踪正在执行过程中的变量或表达式的值。

在设计阶段，选择“调试”|“添加监视”命令或“快速监视”命令来添加监视表达式，设置监视类型，操作步骤如下：

- (1) 选择“调试”|“添加监视”命令，打开“添加监视”对话框，如图 1-2-9 所示。
- (2) 在“表达式”文本框中输入一个变量名或表达式。
- (3) 在“监视类型”选项区域中选中“监视表达式”单选按钮。
- (4) 单击“确定”按钮。

如果将某一表达式设置为监视表达式，则该表达式将出现在“监视”窗口中。按 F8 键运行该程序，连续按 F8 键，此时是用逐语句方式执行程序，窗体上将显示程序的执行情况，同时在“监视”窗口中实时显示变量表达式的值。

值得注意的是，变量是监视表达式最简单的形式，监视表达式还可以是任何表达式，如算术表达式、关系表达式及逻辑

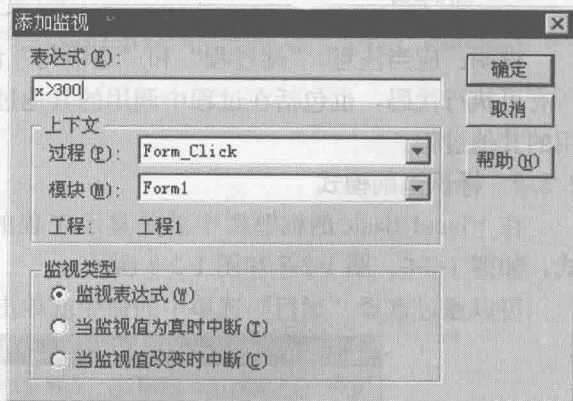


图 1-2-9 “添加监视”对话框