

注重编程思想 激发学习兴趣

Visual Basic.NET

实验指导与编程实例

李印清 等编著



清华大学出版社

Visual Basic .NET 实验指导与编程实例

李印清 等编著

清华大学出版社

北 京

内 容 简 介

本书是与《Visual Basic .NET 程序设计实用教程》配套的辅导教材,全书共4章。第1章介绍了 Visual Basic .NET 程序中可能出现的错误、调试程序的方法、各种断点的设置和高级调试技术;第2章是 Visual Basic .NET 程序设计实验部分,按照 Visual Basic .NET 程序设计各知识点的要求,共安排了27个实验,每个实验包括实验目的、实验内容、基本要求、实验要点和思考题,其中的大多数实验列举了多个题目,为教师组织实验教学提供了较大的选择空间;第3章对2004年4月和9月全国计算机等级考试 Visual Basic 试卷进行了解析;第4章给出了教学信息管理系统和电子相册应用程序两个软件开发实例,介绍了程序开发方法及其实现过程。附录给出了 VB .NET 程序设计实验报告格式,教师可要求学生将完成的有代表性的实验按此格式要求进行总结,并对其成绩进行评定。

本书中的两个编程实例的文档、源代码和可执行文件附带在与本书配套出版的《Visual Basic .NET 程序设计实用教程》一书所附的光盘中,程序都在 Windows XP 平台和 Visual Studio .NET 环境下调试通过并经过严格测试。

本书适用于高等院校非计算机专业或计算机科学与技术、软件工程专业程序设计课程的实验教材,也可供广大软件设计爱好者参考。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

Visual Basic .NET 实验指导与编程实例/李印清等编著. —北京:清华大学出版社,2006.5

ISBN 7-302-12954-1

I. V… II. 李… III. BASIC 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 041903 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦
http://www.tup.com.cn 邮 编:100084
社 总 机:010-62770175 客户服务:010-62776969

责任编辑:刘利民

文稿编辑:鲁秀敏

封面设计:张 岩

版式设计:郑轶文

印 装 者:北京牛山世兴印刷厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:11 字数:234 千字

版 次:2006年5月第1版 2006年5月第1次印刷

书 号:ISBN 7-302-12954-1/TP·8234

印 数:1~5000

定 价:15.00 元

前 言

实践是学习程序设计课程十分重要的教学环节,通过上机调试程序过程的实训,可以加深对编程环境、语法和实现算法的理解与掌握。为此,作者编写了本书,该书是《Visual Basic .NET 程序设计实用教程》一书的配套教材。

本书作者有着多年一线教学的经验。鉴于目前相当多的读者有参加全国计算机等级考试的愿望,以及模仿应用程序开发实例设计自己的应用程序的要求,本书内容分为4章。

第1章为 Visual Basic .NET 调试技术,详细介绍了程序中可能出现的错误类型、程序的调试、各种断点。另外,简单介绍了 Visual Basic .NET 高级调试技术。

第2章为 Visual Basic .NET 程序设计实验,共有27个,每个实验包括实验目的、实验内容、基本要求、实验要点和思考题。其中大多数实验中列举了多个题目,教师在组织实验教学时可根据教学大纲中实验时数的多少从中选择学生必须完成的实验题目。另外,本书的附录给出了 VB .NET 程序设计实验报告格式,教师可让学生参照此格式对完成的有代表性的实验进行总结,并对其成绩进行评定。

第3章为全国计算机等级考试 Visual Basic 真题解析,列举出了2004年4月和9月两套试卷,并对每套试卷进行了解答和解释。最后,还简单介绍了 Visual Basic 与 Visual Basic .NET 的基本区别。

第4章为 Visual Basic .NET 开发实例。这与《Visual Basic .NET 程序设计实用教程》一书的编写风格相一致,即“学好一门课程的关键因素在于您对该领域有着强烈的兴趣!”。因此,本书组织这一章的目的有两个,其一,是将程序设计的学习和训练与自身的兴趣结合起来;其二,“模仿”是人类学习的一种重要形式,读者可以模仿这一实例设计出自己的应用程序。本章用较大篇幅展示了 Visual Basic .NET 程序开发的方法和 Visual Basic .NET 在面向对象编程中的魅力。本章中的实例详细介绍了教学信息管理系统和电子相册应用程序的功能、程序中使用的核心组件,以及应用程序的设计过程和代码。两个实例均可以直接运行。

本书由李印清等编著,并负责全书的总体策划与统稿、定稿工作。其中齐艳珂编写了第1章、第2章的第7和20节以及第3章,于浩杰编写了第2章的第1、2、3、8、9、10节,李印清编写了第2章的第11、12节和附录,赵小雨编写了第2章的第4、5、6、21、22、23、24、25节和第4章的第2节,薛海燕编写了第2章的第13、14、15、16、17、18、19、26、27节和第4章的第1节,李筱宁负责内容简介、前言、目录、第2章第11和12节以及参考文献的录入和整理工作,赵金林和杨建中对第1章和第3章的初稿进行了审阅。于浩杰负责全书的排版工作。

本书在编写过程中得到了许多同志的支持和帮助。承蒙甘勇教授审阅了全部书稿,并

提出了许多宝贵意见和建议。另外，在编写过程中，编著者还参考了大量的文献资料。在此，对甘勇教授和这些资料的作者一并致以衷心的感谢。

由于时间仓促，书中难免有疏漏和欠妥之处，敬请各位专家、读者不吝批评指正。

编著者

2006年5月于熊耳河畔

目 录

第 1 章 Visual Basic .NET 调试技术 1

1.1 程序中的错误类型..... 1	
1.1.1 语法错误..... 1	
1.1.2 运行错误..... 1	
1.1.3 逻辑错误..... 2	
1.2 程序的调试..... 3	
1.2.1 Visual Studio .NET 的工作模式..... 3	
1.2.2 “调试”菜单..... 4	
1.2.3 调试工具栏..... 6	
1.2.4 调试窗口..... 7	
1.3 断点..... 10	
1.3.1 设置断点..... 10	
1.3.2 配置断点..... 11	
1.3.3 微调断点..... 12	
1.3.4 控制断点..... 13	
1.4 高级调试技术..... 14	
1.4.1 调用堆栈..... 14	
1.4.2 异常处理..... 15	

第 2 章 Visual Basic .NET

程序设计实验 17

2.1 集成开发环境实验..... 17	
2.1.1 实验目的..... 17	
2.1.2 实验内容..... 17	
2.2 简单程序设计实验..... 20	
2.2.1 实验目的..... 20	
2.2.2 实验内容..... 20	
2.3 VB .NET 基本语法实验..... 21	
2.3.1 实验目的..... 21	
2.3.2 实验内容..... 21	
2.4 选择结构实验..... 23	
2.4.1 实验目的..... 23	

2.4.2 实验内容..... 23	
2.5 循环结构实验..... 25	
2.5.1 实验目的..... 25	
2.5.2 实验内容..... 26	
2.6 数组的应用实验..... 30	
2.6.1 实验目的..... 30	
2.6.2 实验内容..... 30	
2.7 排序实验..... 34	
2.7.1 实验目的..... 34	
2.7.2 实验内容..... 34	
2.8 简单过程的设计与调用实验..... 36	
2.8.1 实验目的..... 36	
2.8.2 实验内容..... 36	
2.9 过程调用中的参数传递实验..... 38	
2.9.1 实验目的..... 38	
2.9.2 实验内容..... 38	
2.10 递归函数过程实验..... 40	
2.10.1 实验目的..... 40	
2.10.2 实验内容..... 41	
2.11 数据结构实验——线性表..... 42	
2.11.1 实验目的..... 42	
2.11.2 实验内容..... 43	
2.12 数据结构实验——栈和队列..... 45	
2.12.1 实验目的..... 45	
2.12.2 实验内容..... 45	
2.13 窗体、按钮、单选按钮、文本框、 复选框和分组框控件实验..... 47	
2.13.1 实验目的..... 47	
2.13.2 实验内容..... 47	
2.14 列表框和组合框实验..... 51	
2.14.1 实验目的..... 51	
2.14.2 实验内容..... 51	
2.15 图片框实验..... 53	

2.15.1 实验目的.....	53
2.15.2 实验内容.....	53
2.16 时间、日期组件实验.....	55
2.16.1 实验目的.....	55
2.16.2 实验内容.....	56
2.17 对话框实验.....	57
2.17.1 实验目的.....	57
2.17.2 实验内容.....	57
2.18 菜单的设计与实现.....	59
2.18.1 实验目的.....	59
2.18.2 实验内容.....	60
2.19 多文档应用程序的开发.....	63
2.19.1 实验目的.....	63
2.19.2 实验内容.....	63
2.20 面向对象技术实验.....	64
2.20.1 实验目的.....	64
2.20.2 实验内容.....	65
2.21 顺序文件操作实验.....	66
2.21.1 实验目的.....	66
2.21.2 实验内容.....	66
2.22 随机文件操作实验.....	68
2.22.1 实验目的.....	68
2.22.2 实验内容.....	69
2.23 流文件处理实验.....	70
2.23.1 实验目的.....	70
2.23.2 实验内容.....	70
2.24 图形程序设计实验.....	70
2.24.1 实验目的.....	70
2.24.2 实验内容.....	71
2.25 媒体播放实验.....	72
2.25.1 实验目的.....	72
2.25.2 实验内容.....	73
2.26 数据库开发实验.....	73
2.26.1 实验目的.....	73
2.26.2 实验内容.....	74
2.27 网络应用程序开发实验.....	75
2.27.1 实验目的.....	75
2.27.2 实验内容.....	75

第3章 全国计算机等级考试

Visual Basic 真题解析..... 77

3.1 Visual Basic 试卷(2004年4月)	
解析.....	77
3.1.1 试题部分.....	77
3.1.2 试题解析.....	87
3.2 Visual Basic 试卷(2004年9月)	
解析.....	93
3.2.1 试题部分.....	93
3.2.2 试题解析.....	105
3.3 Visual Basic 和 Visual Basic .NET	
的基本区别.....	112

第4章 Visual Basic .NET 编程实例. 114

4.1 教学信息管理系统.....	114
4.1.1 教学信息管理系统简介.....	114
4.1.2 系统中使用的核心组件.....	116
4.1.3 教学信息管理系统的实现过程.....	118
4.1.4 “学生信息”子选项卡的实现.....	119
4.1.5 “课程信息”子选项卡的实现.....	129
4.1.6 “选课信息查询”子选项卡的实现.....	130
4.1.7 “用户管理”子选项卡的实现.....	133
4.1.8 添加用户登录界面.....	134
4.2 电子相册应用程序的设计与开发.....	136
4.2.1 电子相册应用程序的功能.....	136
4.2.2 电子相册应用程序所用到的核心组件.....	137
4.2.3 命名空间和窗体变量.....	138
4.2.4 组件初始化.....	139
4.2.5 新建相册.....	140
4.2.6 打开相册.....	141
4.2.7 删除相册.....	147
4.2.8 关闭相册.....	149
4.2.9 退出.....	150
4.2.10 上一张和下一张.....	150
4.2.11 自动浏览.....	151
4.2.12 添加相片.....	152
4.2.13 另存为.....	153

4.2.14 背景.....	154	4.2.21 工具栏的工具按钮事件	162
4.2.15 扩大和缩小.....	155	附录 Visual Basic .NET 程序设计	
4.2.16 镜像.....	156	试验报告格式	163
4.2.17 透明.....	157	参考文献	165
4.2.18 旋转.....	158		
4.2.19 还原.....	159		
4.2.20 帮助.....	160		

第 1 章

Visual Basic .NET 调试技术

在应用程序的开发过程中，无论是初学者还是拥有多年编程经验的程序员，在编写程序时都很难避免各种错误的发生。发现与消除程序中的错误，即程序调试，是程序设计中必不可少的一个重要环节。只有通过调试才能设计出符合编程人员设计意图和设计功能的应用程序。因此，掌握常用的调试技术是 VB .NET 初学者必须具备的基本技能。

1.1 程序中的错误类型

程序中的错误千差万别，各不相同，然而可以将 VB .NET 程序设计中可能遇到的错误分为语法错误、运行错误和逻辑错误 3 种。

1.1.1 语法错误

语法错误是指程序中出现了违反语法规则的错误。由于编译程序对源程序进行编译时要对源程序进行语法检查，因此若产生此类错误，将导致源程序无法通过编译。所以，对这类错误而言，是比较容易发现和纠正的。

语法错误往往是初学者对语句格式理解不透彻或者错误或者疏忽大意而产生的。例如关键字 If、Select、For、Do 等输入错误，结构性语句中出现结构缺陷，使用了非法的变量名或函数名等。

例如，若编译程序时产生如下提示信息：

```
F:\VB .NET\Visual Studio Projects\HelloWorld\Form1.vb(83)
```

For 必须以匹配的 Next 结束。则表示在应用程序的第 83 行出现语法错误，错误原因是第 83 行的 For 循环语句没有配套的 Next 结束语句。

1.1.2 运行错误

运行一个没有语法错误的程序未必能够得到正确的结果，这是因为程序中还可能会有运行错误存在。所谓运行错误是指在程序的运行过程中，程序代码试图执行无法处理的操

作而触发的错误。引起运行错误的原因各种各样，如除数为 0、对负数开平方、死循环等。此类错误由于程序本身没有语法错误，通常比较隐蔽，需要根据源程序和错误现象仔细推敲，一般只有在程序运行时才能检测到。

积累排除运行错误的经验对初学者来讲是重要的。因此，初学者应该多上机实习，不要惧怕程序出错，实际上，成功排除运行错误的次数越多，程序设计能力提高的就会越快。

【例 1.1】具有运行错误的程序示例。其程序代码段如下：

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As _  
    System.EventArgs) Handles Button1.Click  
    Dim result As Integer  
    Dim i As Integer  
    TextBox1.Text = ""  
    result = 1  
    For i = 1 To 15  
        result = result * i  
        TextBox1.Text += i.ToString() + "!=" + result.ToString() + vbCrLf  
    Next  
End Sub
```

运行程序时出现如下错误信息：

未处理的 **System.OverflowException** 类型的异常出现在 first.exe 中。

其他信息：算术运算导致溢出。

错误的原因在于，在进行乘法运算时，随着数据的不断增大，得到的结果很可能会超过 **Integer** 类型所能表示的数据范围。题目中正是因为在进行计算时，**Integer** 型变量 **result** 中要存放的数据超出了其所能表示的最大数据，而导致了“溢出”。改正的方法是把变量 **result** 的数据类型由 **Integer** 类型更改为 **Long** 类型。

1.1.3 逻辑错误

所谓逻辑错误，指的是应用程序没有语法错误和运行错误，但是运行程序时无法得到正确的结果。导致程序产生逻辑错误的主要原因有：选用的变量类型不正确，导致运行程序时产生精度误差；为变量指定了错误的初始值；逻辑判断条件与题意不符；选择结构中各分支语句写颠倒；表达式书写错误；算法本身存在错误。

由于系统不会显示逻辑错误提示信息，所以查找程序中存在的逻辑错误是比较困难的。有些较大或较复杂的程序，在使用很久之后还可能会发生逻辑错误。

找到程序中存在的逻辑错误的方法是通过单步测试程序对相关变量的取值和每次出现的结果进行分析和查找。

【例 1.2】具有逻辑错误的程序示例。实现两个数据的交换，其运行界面如图 1.1 所示。程序代码如下：

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As _
```

```

System.EventArgs) Handles Button1.Click
    Dim x As Double, y As Double
    x = Val(TextBox1.Text)
    y = Val(TextBox2.Text)
    x = y : y = x '交换数据
    TextBox3.Text = x
    TextBox4.Text = y
End Sub

```

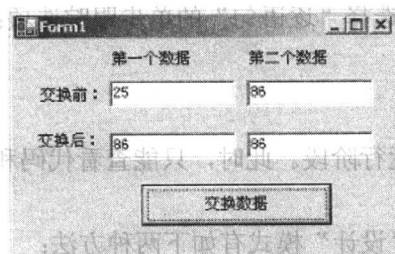


图 1.1 具有逻辑错误的程序界面

首先,程序不存在语法错误,运行程序后,在文本框中输入数据,交换后的结果如图 1.1 所示,产生了错误的结果,即在执行程序过程中丢失了一个数据。分析之后可以发现,赋值语句是把表达式的计算结果保存到赋值号左侧的变量中,存储单元存储信息的特点是“取之不尽,新来旧去”,而赋值时先“擦除”原内存单元中的数据,然后把表达式的值存入其中。

纠正上述错误的方法是使用中间变量,将其中一个变量的值暂存一下。相应的交换两个变量值的赋值语句可写为:

```
t = x : x = y : y = t
```

1.2 程序的调试

在调试应用程序时,如何能够快速、高效地查找其中存在的错误呢?VB .NET 提供了功能强大的调试工具,调试器包括“调试”菜单、调试工具栏、断点和调试窗口。使用这些工具能帮助分析程序运行过程,分析变量和属性是如何随着语句的执行而改变的,以便确定程序代码中存在的错误。

1.2.1 Visual Studio .NET 的工作模式

Visual Studio .NET 提供了“设计”、“运行”和“中断”3 种工作模式。

1. “设计”模式

“设计”模式代表程序处于设计编写代码阶段。在此模式下,用户可以设计窗体、绘

制控件、编写代码, 以及使用属性窗口设置、查看和修改属性值等。但是应注意, 在该模式下只能使用设置断点和向监视窗口添加监视表达式两种调试工具, 不能使用其他工具。

由“设计”模式转换为“运行”模式有 3 种方法:

- (1) 在“调试”菜单中选择“启动”选项;
- (2) 直接单击标准工具栏中的“启动”按钮;
- (3) 直接按功能键 F5。

由“设计”模式转换为“中断”模式有如下两种方法:

- (1) 在“调试”菜单中选择“逐语句”的单步跟踪选项;
- (2) 直接按功能键 F11。

2. “运行”模式

此种模式表明程序处于运行阶段。此时, 只能查看代码和程序的执行结果, 但不能修改程序代码。

由“运行”模式转换为“设计”模式有如下两种方法:

- (1) 在“调试”菜单中选择“停止调试”选项;
- (2) 直接单击调试工具栏中的“停止调试”按钮。

由“运行”模式转换为“中断”模式也有两种方法:

- (1) 在“调试”菜单中选择“全部中断”选项;
- (2) 直接单击调试工具栏中的“全部中断”按钮。

3. “中断”模式

此种模式表示程序处于中止状态或单步跟踪调试状态。在此模式下, 可以使用所有的调试工具, 可以查看程序代码并进行修改。在中止状态下, 还可以根据不同需要决定应该重新启动程序、结束执行还是从中断处继续执行程序。

进入“中断”模式有如下 3 种方法:

- (1) 在“调试”菜单中选择“全部中断”选项;
- (2) 直接单击工具栏中的“全部中断”按钮;
- (3) 直接按组合键 **Ctrl+Alt+Break**。

1.2.2 “调试”菜单

VB .NET 中的“调试”菜单提供了强大的调试功能, 如图 1.2 和图 1.3 所示。下面简单介绍 VB .NET 的“调试”菜单中的菜单项。

1. 断点

该调试方式用于指定程序停止运行的位置。若设置程序中的某一行作为断点, 则运行到此行后即会停止运行, 此时可以方便地查看变量的当前值和中间结果等。

2. 逐语句

该调试方式又称为单步执行, 用于按照语句逐条执行的方式执行整个程序。在该方式

下可以使用“即时”窗口、“局部变量”窗口或“监视”窗口详细地观察程序执行每个语句时的变量值和结果等。注意，如果当前语句是一个过程调用，则下一个被显示的语句是该过程内的第一个语句。

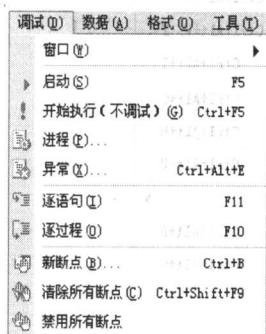


图 1.2 “设计”模式下的“调试”菜单

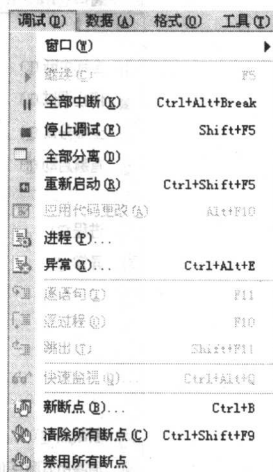


图 1.3 “运行”模式下的“调试”菜单

启动单步执行的最简便方法是按 F8 键。每按一次 F8 键，VB .NET 首先处于运行状态来执行一条语句，并将执行指针指向下一条要执行的语句，然后切换到中断模式。

3. 逐过程

这种调试方式类似于“逐语句单步跟踪”，其区别在于当运行到过程或函数时，不进入过程或函数的内部再单步执行，而是执行一次整个过程，再单步执行后续的语句。

4. 跳出

“跳出”用于中止当前过程的单步跟踪，令当前过程的剩余部分一次执行完，然后再恢复为单步跟踪状态。

5. 停止调试

选用该菜单项的目的在于停止调试应用程序。

6. 重新启动

选用该菜单项的目的在于从应用程序的开始位置执行应用程序。如果应用程序正在运行，程序将会重新启动。

7. 窗口

该菜单项的级联菜单用于控制调试时的特有窗口的显示与否，如自动窗口等。注意，只有当启动了“运行”模式后，该菜单控制的所有窗口才会在子菜单中显示出来，如图 1.4 所示。

8. 局部变量

单击此菜单项将弹出“局部变量”窗口。该窗口用于显示执行当前程序时其中的局部

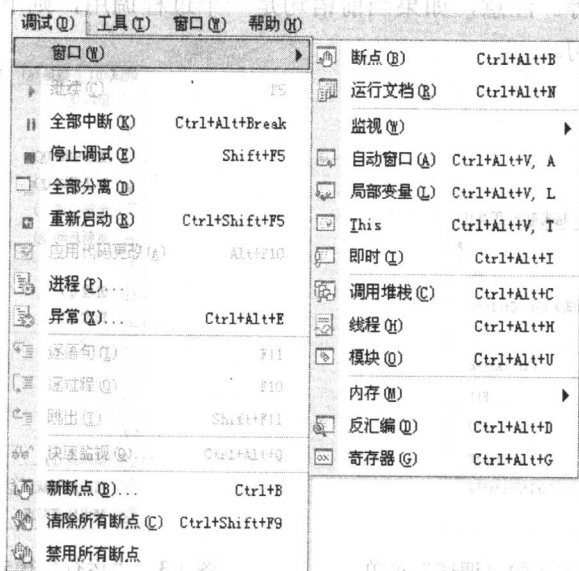
[illegible]

图 1.4 “运行”模式下的“窗口”子菜单

9. 调用堆栈

单击此菜单项将弹出“调用堆栈”窗口。该窗口用于显示所有在当前程序中已经被调用但还未结束运行的过程的列表。

1.2.3 调试工具栏

为了使用户更方便地调试程序，VB .NET 集成开发环境提供了如图 1.5 所示的调试工具栏。其按钮与对应的菜单命令功能相同。需要说明的是，按钮何时起作用与当时的状态有关。例如，在“设计”模式下，只有“启动”按钮、“逐过程”按钮、“逐语句”按钮等可用，“全部中断”按钮、“停止调试”按钮、“重新启动”按钮等不可用；又如，如果采用“逐语句”方式执行程序，则在“中断”模式下只有“全部中断”按钮不可用。

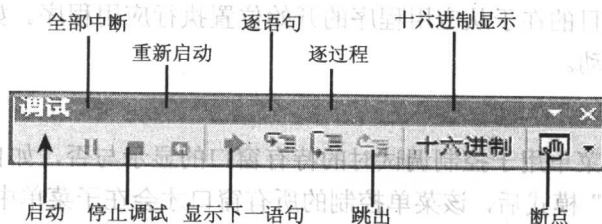


图 1.5 VB.NET 调试工具栏

1.2.4 调试窗口

【例 1.3】编程求 $1+2+3+\dots+15$ 之和，程序界面如图 1.6 所示。

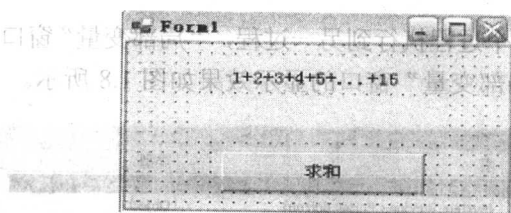


图 1.6 程序界面 (例 1.3)

编写代码如下：

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As _
    System.EventArgs) Handles Button1.Click
    Dim i As Integer, sum As Integer
    sum = 0
    For i = 1 To 15
        sum += i
    Next
    Label1.Text = sum
End Sub
```

以“逐语句”方式执行程序的语句时，可以使用“调试窗口”调试程序。下面结合例 1.3 介绍几个调试窗口的使用方法。

1. “监视”窗口

调试程序时可以利用“监视”窗口显示执行程序时的变量值或表达式值。打开“监视”窗口的方法是，选择“调试”菜单，单击“窗口”命令，在弹出的级联菜单下单击“监视”命令，在弹出的子菜单中任意选择一个，如“监视 1”，然后单击“名称”列最下面一行的单元格，在其中输入要监视的变量或表达式。

VB .NET 共提供了 4 个不同的“监视”窗口，用户可以利用不同的窗口把要观察的变量进行分类。例如，把某段代码中的变量放在“监视 1”窗口，把另一段代码中的另一变量放在“监视 2”窗口，这样可以同时观察程序不同作用域中的变量。

运行例 1.3 程序后，在“监视 1”窗口中输入要监视的变量 sum，其效果如图 1.7 所示。

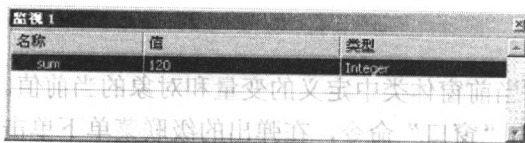


图 1.7 在“监视窗口”中监视 sum 变量的值

2. “局部变量”窗口

“局部变量”窗口用来显示执行当前过程中所有的局部变量值。打开“局部变量”窗口的的方法是，选择“调试”菜单，单击“窗口”命令，在弹出的级联菜单下单击“局部变量”命令。

注意，运行程序时若从一个过程执行到另一过程，“局部变量”窗口的内容会随之改变。运行例 1.3 程序后，“局部变量”窗口的显示效果如图 1.8 所示。

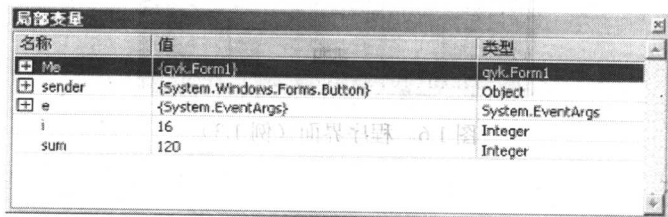


图 1.8 在“局部变量”窗口中监视变量的值

对于复杂的数据类型和对象，左侧显示加号，如图 1.9 中的 e，用户可以展开并访问其所包含的数据成员。

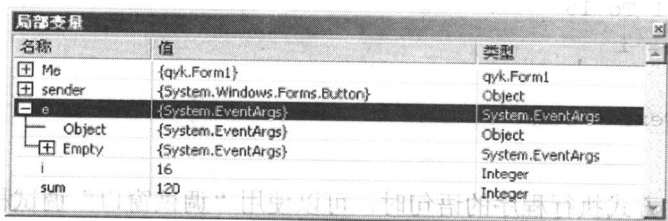


图 1.9 在“局部变量”窗口中监视复杂类型的变量

在“逐语句”调试程序时，用红色标记前一步改变的变量值。双击“局部变量”窗口的“值”列可选中并修改数据成员的值（选中局部变量中一行直接输入数据修改亦可）。这在用户发现了错误并且打算继续调试其余程序代码时非常有用。

3. “自动”窗口

“自动”窗口用来显示执行某过程当前代码行的前、后两行中的变量值。打开“自动窗口”的方法是，选择“调试”菜单，单击“窗口”命令，在弹出的级联菜单下单击“自动窗口”命令。

假设执行例 1.3 程序的当前执行的代码行是设置断点的代码行，“自动”窗口显示如图 1.10 所示。

4. Me 窗口

Me 窗口用来显示在当前窗体类中定义的变量和对象的当前值。打开 Me 窗口的的方法是，选择“调试”菜单，单击“窗口”命令，在弹出的级联菜单下单击 Me 命令。

运行例 1.3 程序后，Me 窗口显示内容如图 1.11 所示。

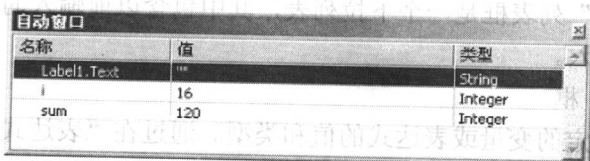


图 1.10 “自动窗口”的显示效果

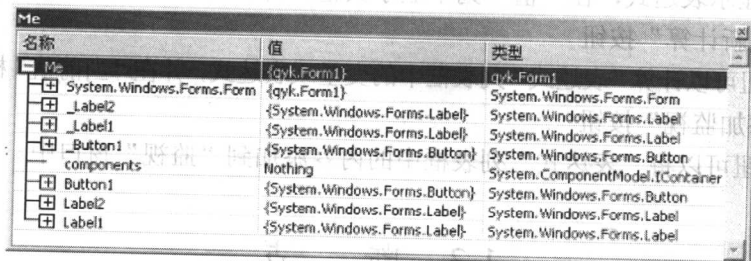


图 1.11 Me 窗口的显示效果

5. “即时”窗口

用户可在“即时”窗口中直接输入操作语句并查看执行结果，如计算表达式、执行语句、打印变量值等。在“即时”窗口中也可以改变变量、对象的值。打开“即时”窗口的方法是，选择“调试”菜单，单击“窗口”命令，在弹出的级联菜单下单击“即时”命令。

还可以在“即时”窗口中输入 Visual Studio .NET 命令。若输入单个 Visual Studio .NET 命令必须以大于号“>”开头。例如，若要从“即时”窗口切换到“命令”窗口，命令必须以大于号“>”开头，即在“即时”窗口中输入“>cmd”。

6. “快速监视”窗口

在进行程序调试时，可以将变量或表达式添加到“监视”窗口，但假如只想快速计算一个或几个变量，而并不想动用“监视”窗口来进行此类计算时，“快速监视”窗口就会助你一臂之力。顾名思义，“快速监视”为查看和计算变量与表达式提供了一个快捷的途径。

“快速监视”是一个模式对话框，必须先关掉它才能继续调试。打开“快速监视”窗口的方法是，选择“调试”菜单，单击“快速监视”命令即可。打开“快速监视”窗口时必须处于“中断”模式。

“快速监视”窗口的另一个好处是它的大小是可变的。如果要查看一个较大对象的成员，在“快速监视”窗口中通常比在“监视”、“局部变量”或“自动窗口”窗口中更方便展开和查看。

将例 1.3 中的语句行 `sum += i` 设置为断点，并在中断状态下打开“快速监视”窗口，如图 1.12 所示。

(1) “表达式”列表框

在该列表框中可以输入变量名或表达式。若要计算表达式的值，将新表达式输入到“当前值”框中并单击“重新计

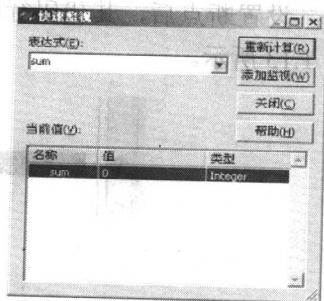


图 1.12 “快速监视”窗口界面