

微软.NET上机实践教程系列丛书

```
Private Sub btnWrite_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles btnWrite.Click  
    Dim worker1 As Worker  
    Dim bwWorker As BinaryWriter  
    Dim fsWorker As FileStream  
    Try  
        If Dir("file2.bin") = "" Then  
            fsWorker = New FileStream("file2.bin", FileMode.OpenOrCreate,  
FileAccess.Write)  
        Else  
            fsWorker = New FileStream("file2.bin", FileMode.Append,  
FileAccess.Write)  
        End If  
        bwWorker = New BinaryWriter(fsWorker)  
        If txtNumber.Text = "" And txtAge.Text = "" And txtWage.Text = "" Then  
            Throw New Exception  
        End If  
    End Try  
End Sub
```

VB.NET

VB.NET

即学即用

上机实践教程

中国民航出版社

王永生等 编 著

VB.NET 上机实践教程

策划：何学仪

主编：王永生

编者：王永生 戴大勇

江苏工业学院图书馆
藏书章

中国民航出版社

①
②
③
④
⑤ 图书在版编目(CIP)数据
⑥
⑦
⑧

VB.NET 上机实践教程/王永生 等编著. —北京:
中国民航出版社, 2002.2
ISBN 7-80110-426-9

I. V...

II. 王...

III. 编程语言-教材

IV. TP391.4

中国版本图书馆 CIP 数据核字(2001)第 057621 号

VB.NET 上机实践教程

王永生 等编著

出版	中国民航出版社
社址	北京市朝阳区光熙门北里甲 31 号楼 (100028)
发行	中国民航出版社
电话	64290477
印刷	中国电影出版社印刷厂
开本	787×1092 1/16
印张	17.5
字数	420 千字
版本	2002 年 2 月第 1 版 2002 年 2 月第 1 次印刷

书号	ISBN 7-80110-426-9/G·124
定价	30.00 元(含光盘)

(如有印装错误, 本社负责调换)

前 言

VB 6.0 的方便好用早已为世人所称道,但随着使用的日益广泛,它的一些不足之处也越来越明显地暴露出来。比如 VB 6.0 不支持类的继承,在与系统进行交互操作时,有时必须调用晦涩难懂的 API 函数等,这些都使 VB 6.0 的使用受到极大的限制。

基于此,微软于 2001 年 9 月发布了 Visual Studio.NET 7.0 的 Beta2 版, Visual Basic.NET (简称 VB.NET) 是其中的一个重要成员。与 VB 6.0 相比, VB.NET 有了本质性的改变,它功能更为强大,使用更为方便。它全面支持类的继承、重载及接口,提供结构化的错误处理及一套全新的语言规范,可以说, VB.NET 是一门真正的面向对象的可视化编程语言。

本书全面介绍了 VB.NET 的概念及应用技术,并提供了大量的实例。本书共有 10 章,从内容上划分,可分为两部分:

第一部分为基础篇,从第 1 章到第 4 章,主要介绍了 VB.NET 集成开发环境、VB.NET 应用程序的界面设计,包括窗体、控件、菜单、工具栏及状态栏的使用。

第二部分为提高篇,从第 5 章到第 10 章,主要介绍了 VB.NET 一些新增的、比较高级的功能。其中第 5 章介绍了类和对象,内容涉及类的创建、类的继承、共享成员、名称空间及接口;第 6 章介绍了文本文件及二进制文件的读写;第 7 章介绍了错误处理语句,其中重点介绍了结构化的错误处理语句;第 8 章介绍了如何使用 ADO.NET 进行数据库的操作;第 9 章介绍了报表的制作;第 10 章介绍了如何创建和调用 Web 服务。

本书由何学仪策划,王永生主编;第一章、第二章、第三章及第九章由戴大勇编写;第四章到第七章由王永生编写;第八章、第十章及附录由王均明编写。此外,在本书的编写和排版过程中,还得到了周敏和许芳等的大力帮助,在此一并表示感谢。

由于 VB.NET 新增的功能太多,而且微软至今尚未发布正式版,笔者对 VB.NET 的认识和理解必然有许多欠妥之处,因而在本书中,错误将在所难免。为此,笔者在网站 <http://www.longliving.net> 上拟加入下列内容:

- ◇ 本书的错误或不足之处的纠正。
- ◇ 针对微软发布的新版本而重写本书部分范例的程序代码。
- ◇ VB.NET 最新的发展动向。
- ◇ VB.NET 新技术的应用举例。

希望这样能够对读者朋友有所帮助,同时也希望大家多多批评指教。

目 录

第一章 VB.NET 概述及编写第一个应用程序.....	1
第二章 窗体和对话框.....	12
2.1 窗体.....	12
2.2 用户自定义对话框.....	20
2.3 预定义对话框.....	23
第三章 标准控件.....	30
3.1 用于显示和输入文本的控件.....	30
3.2 供用户执行动作的控件.....	39
3.3 为用户提供选择项的控件.....	41
3.4 其它一些控件.....	54
第四章 菜单、工具栏、状态栏及多文档界面.....	64
4.1 菜单设计.....	64
4.2 工具栏设计.....	71
4.3 状态栏设计.....	75
4.4 多文档应用程序的开发.....	77
第五章 面向对象程序设计.....	87
5.1 类的创建.....	87
5.2 类的继承.....	104
5.3 共享类成员.....	112
5.4 名称空间(Namespace)	123
5.5 接口(Interface)与实现(Implement)	132
第六章 文件的读写.....	157
6.1 文本文件的读写.....	157
6.2 二进制文件的读写.....	165

第七章 错误处理	175
7.1 非结构化错误处理语句	175
7.2 结构化错误处理语句	180
第八章 使用 ADO.NET 操作数据库	190
8.1 利用 OleDbCommand 对象和 OleDbDataReader 对象读取数据	190
8.2 利用 OleDbDataAdapter 对象和 DataSet 对象来存取数据	200
8.3 利用 OleDbCommand 对象插入、修改或删除记录	213
8.4 普通控件与数据源的绑定	220
8.5 DataGrid 控件与数据源的绑定	223
第九章 报表制作及调用	229
9.1 报表的制作	229
9.2 报表的调用	236
第十章 创建和调用 Web 服务	239
附录 思考与练习参考答案	254

第一章 VB.NET 概述及编写第一个应用程序

本章学习目标

- ◇ 了解 Microsoft .NET
- ◇ 了解 VB.NET 及其新特性
- ◇ 了解 VB.NET 集成开发环境
- ◇ 编写第一个 VB.NET 应用程序

随着 IT 业竞争的日益加剧, 各大软件公司纷纷制定了自己的 .NET 计划, 以期在 IT 业占有一席之地。作为全球软件业的巨子, 微软公司更是不敢怠慢。2000 年 6 月 22 日, 微软对外宣布了它的 Microsoft.NET 战略, 并宣称: “.NET 的意义不亚于从 DOS 改变为 Windows 的过程。 .NET 战略是含义广泛的运算设想, 将改变在因特网上存贮和传送信息的模式、各运算单元相互连接的方式、我们对软件应用的思路, 甚至改变我们与计算机进行交流的方式。 .NET 代表着从 Internet 浏览和静态网站世界到动态和智能数据交换世界的跨越。计算机有史以来首次名副其实地可代表计算机用户进行工作。”

但 .NET 究竟是什么? 微软对此的定义是 “一个建立下一代 Internet 上网的平台”, 如此定义虽然简洁, 但无法让用户明白其确切的含义。通常认为, .NET 就是指 .NET 框架 (.NET FrameWork), 它是一种分布式运算的框架, 以 XML 为基础, 以 Web 服务为核心, 辅以其他多种技术实现, 意在充分利用 Internet 上强大的计算资源和丰富的带宽资源, 提高用户的工作效率。

Microsoft.NET 计划的策略是将互联网本身作为构建新一代操作系统的基础, 对互联网和操作系统的设计思想进行合理延伸。这样, 开发人员必将创建出摆脱设备硬件束缚的应用程序, 以便轻松实现互联网连接。Microsoft.NET 无疑是当今计算机技术通向计算时代的一个非常重要的里程碑, 这同时也表明, 对于当前 IT 业的从业人员来说, 学习并掌握 Microsoft.NET, 对自己在 IT 业的发展有着相当重要的意义。

Visual Studio.NET 是作为 Microsoft.NET 计划中的一个产品与服务推向用户的, 目前的版本为 Beta 2.0, 是一个仍处于测试阶段的 .NET 开发平台。本书所讲的 Visual Basic.NET 则是 Visual Studio.NET 家族的一个成员, 有时为延续版本及称呼上的方便, 又将 Visual Basic.NET 称作 Visual Basic 7.0, 为统一起见, 本书称为 Visual Basic.NET, 简称 VB.NET。

要点概述

1. VB.NET 的新特性

VB.NET 是 VB 的后续版本, 作为微软 .NET 计划中的一个产品, 您可以用它轻松地创建基于 Windows 和 Web 的下一代应用程序。使用 VB.NET, 就会使得 Web 应用程序、Web 服

务、Windows 应用程序和服务端组件的可视化开发变得非常方便。此外，值得提出的是，VB.NET 使用了 Windows 应用程序的 XCOPY 部署方案，用户可以不必再担心 DLL 版本问题。可以说，随着 VB.NET 的发布，“DLL 噩梦”将成为历史。

在设计 VB.NET 时，微软尽量考虑了 VB 开发者的要求。VB.NET 提供了一流的面向对象的程序设计语言特性，诸如实现继承、重载和参数化的构造器。此外，开发人员可以通过显式的自由线程编写具有高度可伸缩性的代码，同时通过其它现代化的语言概念（如结构化的错误处理等）编写具有较高可维护性的代码。VB.NET 将提供开发人员创建强壮、可伸缩的分布式 Web 应用所需要的所有语言特性。

以下简要介绍一下 VB.NET 在语言及其它方面的一些新特性：

(1) 真正的继承性

VB.NET 的新特性中最令人激动的应该是类的继承了，因为一直以来，VB 的开发人员对 VB 特性的要求中呼声最高的就是对类的继承的支持。Internet 时代的开发要求快速地组装和大量的代码重用，而 VB.NET 现在已完全支持继承（包括可视化窗体的继承）。因此，VB.NET 的继承性也可以说是形势使然，众望所归。

(2) 重载

VB.NET 允许重载类的方法，这使得开发人员可以在类中定义多个名称相同但参数列表不同（包括参数的个数、类型及返回值的类型都可以不同）的方法，这样，类的同一个方法就可以适用于不同的情况。

例如，我们定义一个重载的 Display 方法，它可分别适用于 Char、Integer 及 Double 型数据的处理：

```
Overloads Sub Display (Byval theChar As Char)
...
Overloads Sub Display (Byval theInteger As Integer)
...
Overloads Sub Display (Byval theDouble As Double)
...
```

如果没有重载，就需要为每个过程使用不同的名称或使用 Variant 参数。

(3) 多态性

多态性允许定义同名但不同数据类型的属性或方法。多态性过程允许以单个通用过程为不同数据的处理提供多种实现。

只要进行了一般形式的函数调用，多态性就会保证把函数的实现细节留给接受函数调用的对象。一般情况下，函数调用时用到的参数决定了函数的行为。利用传送的不同参数，可以使同样的对象具有不同的性质、功能。多态性有时也指方法的重载。

(4) 共享成员

共享成员是指由类的所有实例所共享的数据和方法。在类的所有实例中共享一个数据成员或方法是使用继承的 VB.NET 应用程序所需要的。一个共享数据成员独立存在于类的每个实例中。共享方法不同于普通的方法，它并不是隐式的传递类的一个实例。由于这个原因，在共享方法中对非共享数据成员的无限制引用是不允许的。共有的共享成员可以被间接地访问，而且它们可以从类的实例后期绑定。

(5) 参数化的构造器

参数化的构造器（或简称为“构造器”）允许创建一个类的新实例的同时向这个实例传递参数。构造器对于面向对象的编程来说是必需的，因为它允许用户定义的构造代码通过实例的创建者传递参数。

(6) 结构化的错误处理

开发企业级的应用程序要求创建可重用的、可维护的部件。在过去的 VB 版本中，一个具有争议的话题就是其对错误处理的支持。开发人员已经发现一致的错误处理方案意味着大量的赋值代码。利用 On Error Goto 语句的错误处理方法有时会减缓大规模应用程序的开发和维护。其叫法就反映出这样一些问题：如 Goto 所意味的，当一个错误发生时，控制权转移到子程序中一个有标记的位置。一旦错误代码运行，它必须时常通过另外的清除位置来转向，而后者又要经过一个标准的 Goto，最后还是要通过其他的 Goto 或 Exit 来退出过程。使用 Resume 和 Next 的多种组合来处理几个不同的错误将会产生难以读懂的代码，并且在执行路径没有被完全考虑到的时候会导致频繁的错误。

利用 Try...Catch...Finally 这种结构化的错误处理语句，这些问题就不复存在。开发人员还可以嵌套错误处理；同时这是一种用于编写在正常条件和异常条件下执行清洁代码的控制结构。

(7) 自由线程

现在当开发人员在 VB 中创建应用程序时，他们编写的代码是同步的。这意味着每行代码要在下一行代码之前执行。而在开发 Web 应用程序时，可伸缩性是一个必须考虑的问题，开发人员需要一个使并行处理成为可能的工具。

通过自由线程，开发人员可以生成一个线程来完成一些运行时间长的任务、执行一个复杂的查询或运行一个多部分的计算，而应用程序的其它部分继续执行，这样就提供了异步处理。

(8) 严格的类型检查

以前的 VB 语言在可能产生隐式的类型强制转换上是非常自由的。对于赋值和除了引用方式之外的参数传递，VB 编译器可以允许几乎任何一种数据类型通过发生运行时的强制类型转换向其他数据类型转换。如果被转换的数据类型不允许任何的数据丢失，那么使用运行时的强制类型转换将会失败。使用新的附加的编译选项，VB 会针对任何有可能产生运行时错误的类型转换产生编译错误。对于那些可能在运行时产生错误的类型转换，使用 Strict 选项将会使 VB 产生编译错误以改善类型安全性。这些错误包含一些用户无法预料的错误，如数字和字符串之间的自动转换等等。

(9) 初始化设置

VB.NET 支持在变量的声明行中对其初始化。初始化设置可以在包括控制结构的任何地方使用。含有初始化设置的过程级声明的语义与一个声明语句后紧跟一个赋值语句是相同的。换句话说，语句：

```
Dim X As Integer=1
```

与语句：

```
Dim X As Integer  
X=1
```

的作用是等同的。

(10) 新的数据类型

VB.NET 不仅提供许多面向对象的特性，它还引入了新的数据类型：

Char——无符号 16 位整数，用于存放 Unicode 字符。

Short——有符号 16 位整数。在早期 VB 版本中，称为 Integer。

Decimal——96 位十进制值。在早期 VB 版本中，此数据类型只有在 Variant 中使用。

(11) 封装 API

VB 6.0 无疑是优秀的编程工具之一，但在涉及 Windows 32 位系统的核心编程方面，如一些高级功能的实现上，它就显得有些力不从心，就需要调用 Windows API 函数来弥补（图 1-1 中的代码示范了某 API 函数的调用方法）。

```
Private Declare Function AdjustTokenPrivileges Lib "advapi32" ( _  
    ByVal TokenHandle As Long, ByVal DisableAllPrivileges As Long, _  
    NewState As TOKEN_PRIVILEGES, ByVal BufferLength As Long, _  
    PreviousState As TOKEN_PRIVILEGES, ReturnLength As Long) As Long
```

图 1-1

如此复杂的调用方式常常使许多人望而却步，VB.NET 的封装 API 特性使得我们不必再去学习并研究 API 体系结构。

(12) Web 服务（Web Services）

Web 服务是微软.NET 战略中非常重要的一个概念。它的目的是将 Web 站点转变为集组织、应用、服务以及设备于一体的可设计 Web 站点，使 Web 站点不再处于被动的地位。就核心而言，一个 Web 服务就是一个通过标准的网络协议发布的中间层的事件处理函数。既然它们使用 HTTP 作为传送机制，所以只要合适地分配 URL，你就可以简单地在一个网络应用程序中构造多种 Web 服务。在程序运行时，所有这些内部构件之间的调用都会自动打包，通过 XML 接口进行调用。开发者可以在任何平台上、使用任何语言编写和使用 Web 服务。

为了方便 Web 服务的开发，VB.NET 引入了一个新的对象类型，即 Web Service。你可以像创建一个当地的 DLL 文件一样将你的 Web 服务发布到远程服务器上。

(13) 内存的垃圾回收（Garbage Collection）功能

当操作系统（.NET 框架）认为该清除某个对象时就会运行垃圾回收程序。在 VB 6.0 中，当你将一个对象设置为 Nothing 时，它就立刻从内存消失。但是在 VB.NET 中，这并不是马上发生的。在被垃圾回收器终止之前，你的对象还会存活几分钟，所以你总是不能特别肯定类的终止代码是什么时候运行的。虽然这看起来有点奇怪，但是这种非决定性的终结确实有其优势，例如自动地纠正循环引用。

VB.NET 现在已成为一流的面向对象编程语言。从用户的角度来讲，VB.NET 突出了软件的易用性和智能化，无论学习和使用都更容易，这可能是微软的开发工具普及率较高的原因之一；基于 Internet 战略的 VB.NET 设计了更强大的 XML、HTML、Web Form 和 Web 服务的开发环境。因此，我们可以说，VB.NET 已不仅仅是一种语言的简单升级，它已经成为一种新的语言。

2. VB.NET 集成开发环境

VB.NET 的集成开发环境实际上不是 VB.NET 专用的，VC++.NET、C#.NET 及 VB.NET 都使用同一个集成开发环境，这就是 VS.NET 的集成开发环境（集成开发环境简称 IDE，是

Integrated Development Environment 缩写)。VS.NET 使用同一个 IDE 为开发人员提供了极大的方便,这也是微软.NET 计划中的一个目标。VS.NET 的 IDE 具有包括源代码创建、资源编辑、编译、链接以及调试等在内的一系列功能。

如果你第一次启动 VS.NET (如图 1-2 所示),你会发现它的界面似曾相识,因为设计这个 IDE 的工作组以前就曾设计过 VB 的 IDE, VS.NET IDE 的改进建立在设计 VB IDE 的经验之上。

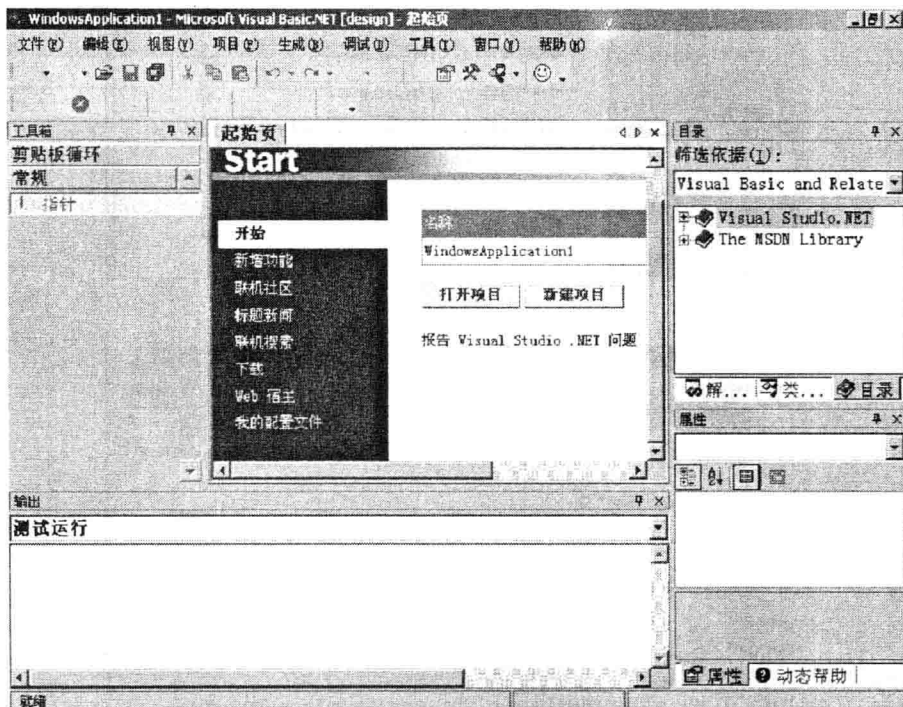


图 1-2

应当说, VS.NET 的界面是相当智能化的,而且感觉上非常宽容,鼓励用户去实践和尝试。下面简单介绍一下 VS.NET 的 IDE 中比较有特色的地方:

(1) 起始页

启动 VB.NET 时,“起始页”是我们看到的第一个界面(如图 1-2 的中间窗口)。“起始页”中显示了最近修改的项目,并提供了打开现有项目、新建项目等选项。所有这些选项都以 HTML 链接的形式出现。需要注意的,这并不是什么模拟出来的效果,“起始页”事实上就是由 DHTML 编写而成。由于这个原因,用户可以定制“起始页”显示其他对自己有用的选项、信息以及 HTML 链接。

用户在使用 VB.NET 之前,最好先设置一下“起始页”中的“我的配置文件”,以便更好地使用 VB.NET。“我的配置文件”设置界面如图 1-3 所示,由于我们将使用 Visual Basic 进行学习、开发,因此将这些参数都设定为 Visual Basic 设置。

(2) 新建项目

单击起始页上的“新建项目”按钮,屏幕上就会出现“新建项目”对话框,在该对话框上提供了用多种语言创建项目的选项,如图 1-4 所示。注意图中左边树形列表中列出了好几种语言。如果系统中已经安装了支持 VS.NET 的第三方语言,Microsoft 或许还会在这个列表

中包含这些语言。

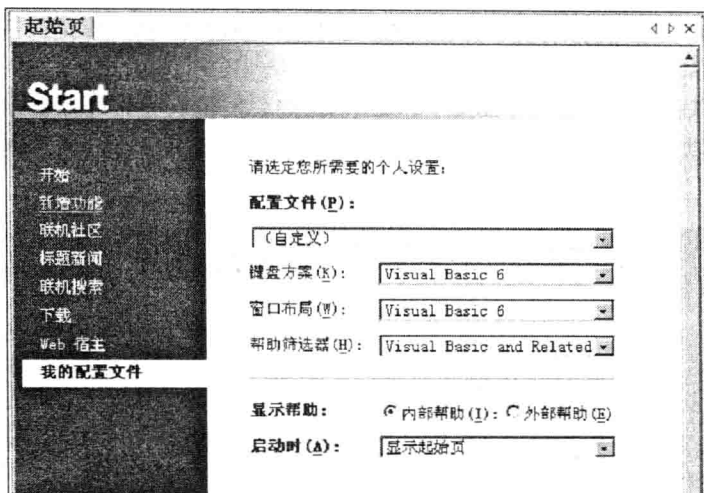


图 1-3

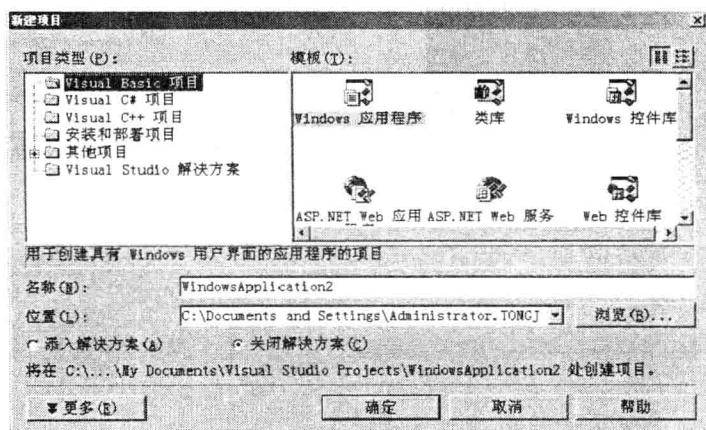


图 1-4

在图 1-4 中，树形列表中的 VB 节点处于打开状态，右边的一些图标显示了创建新项目的几个选项。和以前版本的 VB 相比，这里出现了一些新的选项，如 Web 服务、Web 控件库、类库等。

(3) 自动隐藏

在 VS.NET 的 IDE 中，还有一个值得提及的新特性，这就是“自动隐藏”。在默认情况下，VS.NET 中的大多数工具窗口总是显示在屏幕上，而当屏幕上的窗口非常多时，用户就会感觉很乱。VS.NET 的所有工具窗口都提供了“自动隐藏”按钮，以工具箱窗口为例，如图 1-5 所示。



图 1-5

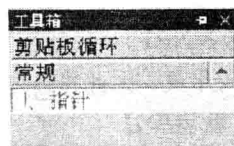


图 1-6

单击“自动隐藏”按钮后，窗口变成如图 1-6 所示的样子。

读者可以注意图 1-6 与图 1-5 的区别：“自动隐藏”按钮改变了显示方向（即从“竖向”变为“横向”）。这时，窗口已实现自动隐藏功能。当用户将鼠标脱离工具箱窗口范围，“自动隐藏”将窗口最小化为图标状，并将其排列在 IDE 的边缘（如图 1-7 所示），从而节省了宝贵的屏幕空间，增大编辑窗口的可视范围。如果用户需要激活该窗口，则只需将鼠标指针指向该窗口的图标，窗口就会自动弹出。



图 1-7

(4) 命令窗口

命令窗口如图 1-8 所示，它是命令行和 VB 即时窗口的结合，因而也具有两个模式：命令模式和即时模式。在命令模式下，用户能够在角括号“>”后面输入 IDE 命令名。为了方便地执行使用频率较高的命令，还可以创建命令的别名或短名。使用命令窗口的目的是，用户可以通过命令窗口执行代码行，对变量进行设置或赋值，或者计算表达式的值。

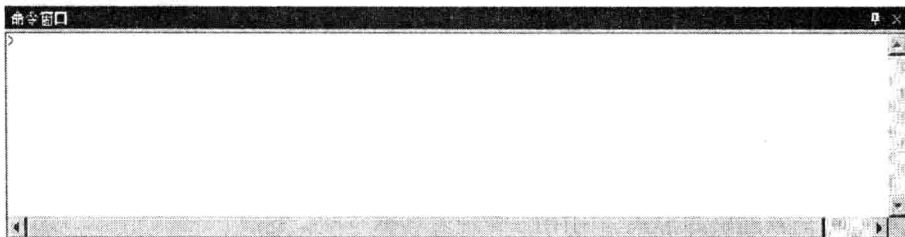


图 1-8

(5) 选项卡式窗口

在进行项目设计中的窗体编辑时，VS.NET 不再使用以前版本中的 MDI 窗口布局，而是采用了选项卡式窗口布局（如图 1-9 所示）。这样开发人员在选择其他窗口时，只需单击相应的选项卡即可，较之以前的 VB 版本中通过“窗口”菜单来选择，要方便得多。

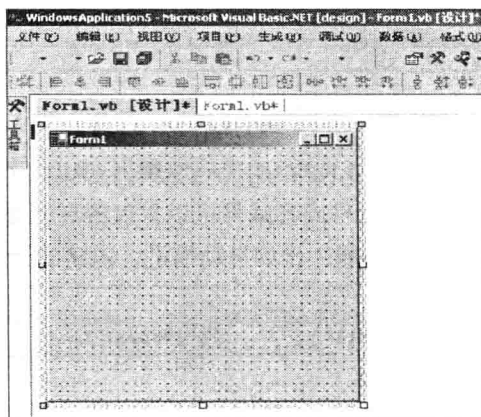


图 1-9

(6) 任务列表

任务列表如图 1-10 所示。和 VB 6.0 相比，“任务列表”是一个重要的改进，它帮助开发

者跟踪所有待完成的任务。代码生成器、编译器会在“任务列表”窗口中自动加上一些内容。手工添加任务项目可以按照如下方式进行：找到代码中需要修改的地方，然后插入一行以“TODO:”开头的注释，此后该任务就会自动出现在任务列表中。点击“任务列表”窗口中的某个任务，代码编辑器中就会出现该任务所处的位置。这种快速定义任务、返回任务的机制将极大地提高程序开发效率。

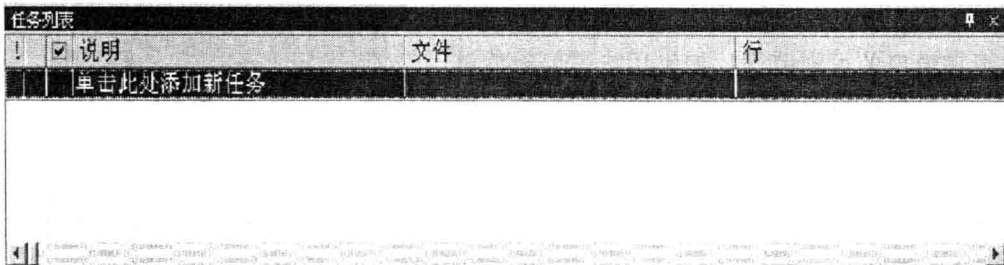


图 1-10

(7) 解决方案资源管理器

“解决方案资源管理器”的外观如图 1-11 所示。它替代了旧版本中的“工程资源管理器”，与 Visual InterDev 中的资源管理器非常相似。和 InterDev 中的资源管理器一样，“解决方案资源管理器”显示了项目所用到的各种资源，包括用其他语言编写的代码模块、各种非代码的资源等。其中“引用”是“解决方案资源管理器”中一种全新的资源形式，“引用”管理着名称空间，类似于 VB 6.0 中的组件引用。

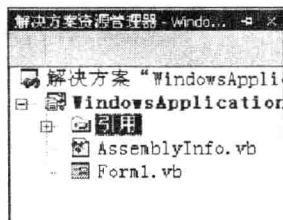


图 1-11

(8) 窗体布局工具条

“窗体布局工具条”的外观如图 1-12 所示。在 VB 6.0 中，在窗体上排列控件的任务通过一个名为“Form Editor”的工具条完成。改进后的新工具条提供更多的功能，可以在用户设计窗体时提供有效的控制。窗体布局工具条在“Windows 应用程序”类型的项目中默认打开。

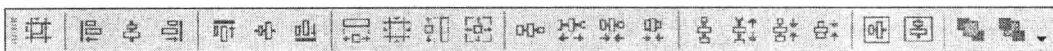


图 1-12

除了以上几处较有特色之外，VB.NET 的代码编辑器还提供了一个重要的新功能：在代码编辑窗口中，单击代码行左边的加号或者减号可以扩展或者折叠代码块，如图 1-13 所示。代码编辑器还有其他一些新的实用功能，比如自动缩进、内建的行号显示功能等。

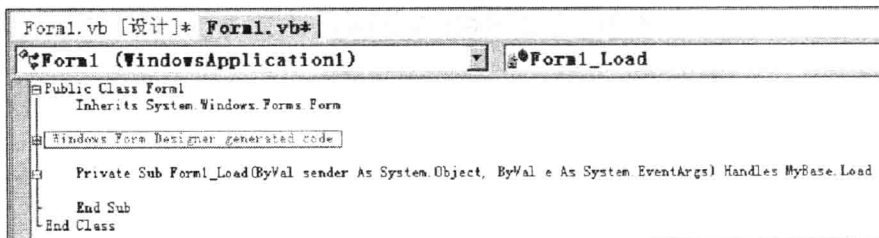


图 1-13

Web 项目的设计环境和原来 VB 6.0 的设计环境略有不同,如图 1-14 所示。在表单布局窗口的下面显示了两个选项卡,分别为“设计”和“HTML”。这两个选项卡的功能和 Visual InterDev 中同一位置的选项卡相似。“设计”选项卡用来编辑窗体的外观,HTML 选项卡用于查看 DHTML 代码。

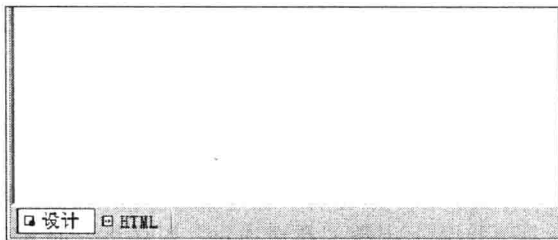


图 1-14

VB.NET 还有一个引人注目的变化是它的窗体引擎。在所有基于 CLR (通用语言运行库, Common Language Runtime) 的 .NET 语言中,微软都放弃了原来的窗体引擎,而使用 Windows Forms 窗体引擎。与 VB 6.0 的窗体引擎相比,Windows Forms 有着许多突出的优点。例如,Windows Forms 引擎支持创建那些自动改变组件大小的窗体、允许把控件锚定到窗体的特定位置等等。也就是说,完成这些特殊任务时再也不需要第三方的工具了。Windows Forms 还有一些有趣的技术,比如创建透明窗体等。

以前的 VB 版本都隐藏了窗体构造过程中的所有细节,用户在 IDE 中设计窗体,然后把代码加入到 Initialize 事件,但对这两者之间的过程却无法进行任何控制。在 VB.NET 中,窗体成了类,类里面包含了构造窗体的全部代码。当然一般情况下,大多数程序员是不会去看这种描述窗口构造细节的代码,但另一方面,由于这些代码允许用户深入控制 VB.NET 构造窗体的幕后细节,有经验的用户就能够利用这些代码实现一些更高级的功能。而如果你不想看到这些代码,你就可以不看,因为新的代码编辑器有扩展和折叠代码区域的功能,描述窗体构造过程细节的代码默认是处于折叠状态的。

3. 创建 VB.NET 应用程序的过程

创建 VB.NET 应用程序有 4 个主要步骤:

- ✧ 创建项目
- ✧ 添加及设置对象属性
- ✧ 编写功能代码
- ✧ 保存并运行程序

本章将通过一个具体的实例来说明创建 VB.NET 应用程序的基本步骤。

上机实践

注意: 本章介绍的所有程序均须保存到 C:\vb.net\ch01 目录下。

【实验:创建 VB.NET 的第一个应用程序】

新建一个项目,在项目中创建一个如图 1-15 所示的

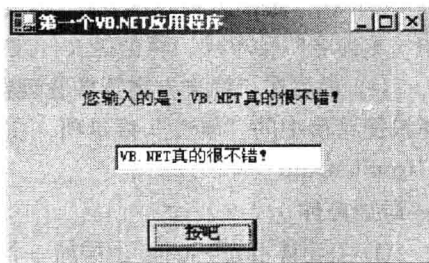


图 1-15

窗体。

程序运行时，在文本框内输入文字或不输入文字，单击按钮“按吧”，就会有不同的结果显示在窗体上：

- ◇ 输入文字时，窗体中标签显示为“您输入的是:”字符串与输入文字相结合的内容；
- ◇ 不输入任何内容时，窗体中标签则显示“对不起，您没有输入内容”。

【操作步骤】

1. 新建项目

(1) 启动 VS.NET，在“起始页”上单击“新建项目”按钮或选择系统菜单中的“文件”/“新建...”/“项目...”项，屏幕上出现“新建项目”对话框。在窗口左侧的“项目类型”列表中选中“Visual Basic 项目”，在窗口右侧的“模板”列表中选中“Windows 应用程序”图标，然后在“名称”文本框中输入项目的名称“prjFirstApp”，在“位置”文本框中输入项目存储的位置“c:\vb.net\ch01”，如图 1-16 所示，然后单击“确定”按钮。

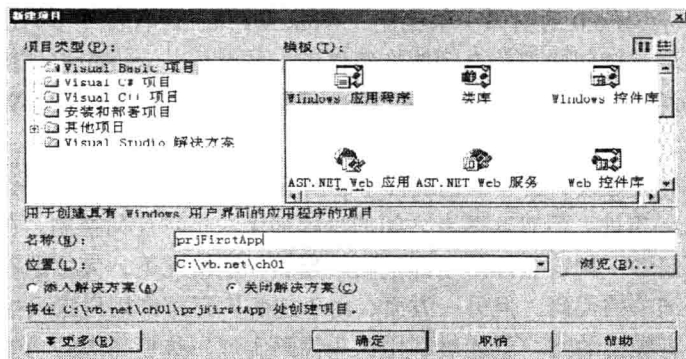


图 1-16

(2) 选中系统自动添加的窗体 Form1，按表 1-1 来设置它的属性。

表 1-1

属性名称	属性值
Name	frmFirstApp
StartPosition	CenterScreen
Text	第一个 VB.NET 应用程序
Size	280, 216

(3) 在“解决方案资源管理器”窗口中，用鼠标右键单击“Form1.vb”，选择快捷菜单中的“重命名”菜单项，将窗体文件改名为“frmFirstApp.vb”。

(4) 最后在“解决方案资源管理器”窗口的项目名称“prjFirstApp”上单击鼠标右键，选择快捷菜单中的“属性”菜单项，在“prjFirstApp 属性页”对话框中将项目的启动对象改成“frmFirstApp”。

2. 添加控件

(1) 在窗体 frmFirstApp 上添加一个标签控件，将 Name 属性的值设置为 lblName，将 AutoSize 属性的值设置为“True”。

(2) 在窗体 frmFirstApp 上添加一个文本框控件，将 Name 属性的值设置为 txtName，将 Text

属性的值设置为空值。

(3) 在窗体 frmFirstApp 上添加一个按钮控件,将 Name 属性的值设置为 butName,将 Text 属性的值设置为“按吧”。此时窗体 frmFirstApp 的外观如图 1-17 所示。

3. 编写代码

在窗体上双击控件 butName (即“按吧”按钮),屏幕上出现 frmFirstApp 窗体的代码编辑窗口,如图 1-18 所示。



图 1-17

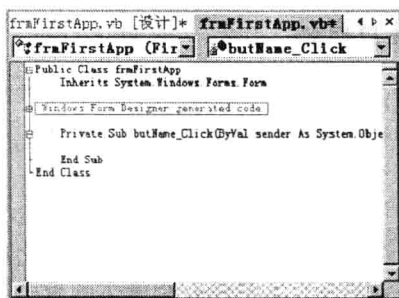


图 1-18

在图 1-18 中所示的 butName_Click 过程中,输入以下代码:

```
If txtName.Text = "" Then
    lblName.Text = "对不起, 您没有输入内容"
Else
    lblName.Text = "您输入的是: " + txtName.Text
End If
```

这段代码的作用是,判断文本框中是否输入了内容,并根据判断的结果在标签控件上显示不同的内容。

4. 保存并运行程序

单击“文件”菜单中的“全部保存”命令,将项目 prjFirstApp 中的所有文件存盘。然后单击“调试”菜单中的“启动”命令,或按 F5 键,运行程序,通过在文本框中输入内容或不输入内容,测试程序是否达到预期目的。

思考与练习

1. Microsoft.NET 的基本含义是什么?
2. VB.NET 提供哪些新特性?
3. VB.NET 的 IDE 有哪些新特性?
4. 编写 VB.NET 应用程序的基本步骤是什么?

1063431