

158

TP312BA
P35C

Mastering Visual Basic .NET

Visual Basic .NET从入门到精通

[美] Evangelos Petroustos 著

邱仲潘 等译

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 提 要

.NET是Microsoft公司在快速发展的技术世界中维持领先的保证, Visual Basic.NET则是生成.NET应用程序的语言, 并且真正将VB转变为强大的面向对象语言。本书结合大量的实例提供了用Visual Basic.NET开发应用程序所涉及原理和技术的坚实基础。首先介绍了Visual Basic.NET基础以及如何建立和使用对象; 接着介绍了.NET的核心——Framework——中的常用类; 然后是中高级课题——图形、调试和递归编程; 最后介绍了数据访问工具和运用ASP.NET开发Web应用程序。本书内容既有深度又有广度, 是编程人员的最佳选择。



Copyright©2002 SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501.
World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

本书英文版由美国SYBEX公司出版, SYBEX公司已将中文版独家版权授予中国电子工业出版社及北京美迪亚电子信息有限公司。未经许可, 不得以任何形式和手段复制或抄袭本书内容。

版权贸易合同登记号: 01-2001-5174

图书在版编目 (CIP) 数据

Visual Basic .NET从入门到精通/ (美) 佩特罗索斯 (Petroustos, E.) 著; 邱仲潘等译. —北京: 电子工业出版社, 2002.7

书名原文: Mastering Visual Basic .NET

ISBN 7-5053-7708-6

I. V… II. ①佩… ②邱… III. BASIC语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字 (2002) 第039173号

责任编辑: 春 丽 叶皓彤

印 刷: 北京天竺颖华印刷厂

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路173信箱 邮编: 100036

北京市海淀区翠微东里甲2号 邮编: 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 59 字数: 1500千字

版 次: 2002年7月第1版 2002年7月第1次印刷

定 价: 88.00元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换, 若书店售缺, 请与本社发行部联系。联系电话: (010) 68279077

目 录

第一部分 基础	1
第1章 VB.NET初步	1
集成开发环境	1
第一个VB应用程序	8
IDE组件	14
环境选项	20
常见属性	21
常见事件	22
常见方法	22
建立控制台应用程序	23
小结	25
第2章 Visual Basic项目	26
建立贷款计算器	26
建立数学计算器	35
将LoanCalculator放到Web上	46
多窗体应用程序	50
发布应用程序	56
小结	63
第3章 Visual Basic语言	65
变量	65
常量	100
数组	101
变量对象	108
流程控制语句	113
小结	124
第4章 编写与使用过程	125
模块化编码	125
变元	133
小结	153
第5章 处理窗体	154
窗体的外观	154
装入与显示窗体	172
设计菜单	182

	运行时建立动态窗体	193
	小结	200
第6章	基本Windows控件	201
	文本框控件	201
	列表框控件、检查表框控件与组合框控件	220
	滚动条控件和跟踪条控件	234
	小结	240
第7章	其他Windows控件	241
	公用对话框控件	241
	RichTextBox控件	255
	小结	273
第二部分 对象		275
第8章	建立定制类	275
	何谓类	276
	建立Minimal类	277
	实际类	299
	继承	309
	多态	313
	谁继承什么	321
	小结	328
第9章	建立定制Windows控件	329
	关于设计Windows控件	329
	改进现有控件	330
	建立复合控件	337
	建立用户绘制控件	340
	设计不规则形状控件	355
	建立自画控件	358
	使用ActiveX控件	363
	小结	364
第10章	Microsoft Office应用程序自动化	366
	编程Word	366
	编程Excel	383
	编程Outlook	392
	小结	404
第三部分 基本Framework类		405
第11章	用集合存储数据	405
	高级数组问题	406

ArrayList集合	413
HashTable集合	420
SortedList类	426
IEnumerator与IComparer接口	430
Serialization类	442
小结	448
第12章 处理字符串、字符与日期	449
处理字符与字符串	449
处理日期	469
小结	483
第13章 使用文件夹与文件	484
访问文件夹与文件	484
访问文件	506
FileSystemWatcher组件	520
小结	524
第四部分 中级编程	525
第14章 Visual Basic图形与绘图	525
显示图形	526
GDI+绘图	535
坐标变换	566
位图	578
小结	592
第15章 VB.NET打印	593
打印对象	593
打印机与页面属性	599
打印举例	611
小结	628
第16章 TreeView与ListView控件	629
高级控件	629
使用TreeView控件	634
使用ListView控件	653
将树节点存盘	665
小结	670
第17章 错误处理与调试	672
错误类型	672
异常与结构化异常处理	676
调试	682
小结	687

第18章	递归编程	688
	基本概念	688
	递归扫描文件夹	695
	堆栈机制	703
	递归编程中的特殊问题	705
	小结	708
第19章	多文档界面	709
	MDI应用程序基础	710
	实现滚动图形框	728
	小结	732
第五部分 VB.NET与数据库编程		733
第20章	数据库体系结构与基本概念	733
	何谓数据库	733
	Server Explorer	740
	结构化查询语言	749
	Query Builder	764
	存储过程	773
	小结	779
第21章	用ADO.NET建立数据库应用程序	780
	ADO.NET体系结构	781
	生成DataSet	782
	数据约束	794
	编程DataAdapter对象	802
	Command与DataReader对象	807
	小结	810
第22章	编程ADO.NET对象	812
	DataSet结构	812
	DataForm向导	817
	事务	824
	进行更新操作	828
	建立与使用定制DataSet	836
	小结	839
第六部分 VB.NET与Web		841
第23章	Web编程简介	841
	HTML启蒙	841
	建立Web应用程序	860
	ASP.NET对象	874

	使用Cookies	877
	在Web应用程序中处理多个窗体	879
	小结	881
第24章	访问Web数据	882
	数据约束Web控件	882
	从Web上取订单	898
	主/细节Web页面	908
	小结	912
第25章	XML Web服务	913
	如何为Web服务	913
	建立Web服务	914
	ASP.NET Web Service项目	918
	数据驱动Web服务	922
	小结	927

第一部分 基础

第1章 VB.NET初步

欢迎使用Visual Basic .NET企业版。这里假设你安装了Visual Basic .NET企业版。也许你已经自己探索了这个新环境，但本书不需要任何Visual Basic 6知识，只要熟悉编程就可以了。众所周知，Visual Basic .NET只是可通过Visual Studio .NET建立应用程序的语言。我坚信这是最简单、最方便的语言，但这并不重要。只要记住，Visual Basic .NET是建立、测试与调试各种应用程序的集成环境：可以建立、测试与调试Windows应用程序、Web应用程序、类与定制控件，甚至包括控制台应用程序。Visual Studio .NET提供了各种工具，可以使开发过程自动化，提供了进行许多常见设计与编程工作的图形化工具，还有许多其他特性。

首先要熟悉这个工作环境。本书第1章介绍集成开发环境（IDE，integrated development environment）和如何用IDE工具迅速设计应用程序用户界面，以及如何编写应用程序。

需要花点时间研究IDE中的所有项目。Visual Studio环境可以开发各种不同的应用程序，包括简单Windows应用程序和完整的Web应用程序（涉及数据库和XML文件）。我准备介绍书中需要的各种项目。本章介绍建立简单Windows应用程序所需的基本IDE组件。

集成开发环境

Visual Studio .NET是开发Windows与Web应用程序的环境。Visual Basic .NET只是编制应用程序的一种语言。实际上，Visual Studio .NET中可以放置任何语言，许多公司开发的语言都可以放进Visual Studio .NET中。有些人在Visual Studio .NET中用COBOL或FORTRAN开发Windows应用程序。

那么，Visual Studio .NET与语言之间有什么区别呢？Visual Studio .NET是个环境，提供开发应用程序所需的所有工具。而语言只是Windows应用程序的一个方面，应用程序的用户界面并不与特定语言相关联，开发应用程序界面的工具也可以让所有编程人员使用，不管他们用什么语言编制应用程序。

访问数据库的工具也是独立于语言的。Visual Studio提供的工具可以连接数据库、插入对象、取得感兴趣的信息和将信息存放在对象中，以便从任何语言中访问。

IDE中有许多图形化工具，如Menu Designer。使用这个工具可以直观地设计菜单和设置菜单名称与基本属性（如选择、启用或关闭某些选项）。设计菜单不需要任何编码，可以用指向与单击操作完成。当然，菜单命令需要插入一些代码，这也可以用任何语言进行。

为了简化应用程序开发过程，Visual Studio .NET提供了所有语言公用的环境，称为集成开发环境（IDE）。IDE的目的是让开发人员尽量利用图形化工具减少编码工作。IDE提供了设计、执行与调试应用程序的工具，是你的第二桌面，你的许多工作都是在IDE环境中完成的。

启动页面

首次运行Visual Studio时，出现图1.1所示的窗口。在My Profile标签中可以指定语言，设置个人选项。选择Profile框中的Visual Basic Developer，另外两个框自动填好。可以让另外的字段保留默认值。页面底部的ComboBox控件是At Startup控件，定义Visual Studio .NET启动时的功能。选项包括：

Show Start Page 每次启动Visual Studio .NET时出现这个页面。

Load Last Loaded Solution 开始处理实际项目之后（项目完成可能需要几天或几个月），选择这个选项以便在每次启动Visual Studio .NET时自动装入这个项目。

Show Open Project Dialog Box 每次启动Visual Studio .NET时打开Open Project对话框，可以选择要打开的项目。

Show New Project Dialog Box 每次启动Visual Studio .NET时打开New Project对话框，可以指定新项目名（应避免这个设置）。

Show Empty Environment 这个选项让Visual Studio .NET启动新的空方案，你要在方案中建立新项目或增加现有项目，在项目中增加新的或现有的内容。

这些操作是一目了然的，最常见的设置是显示启动页面。启动页面显示四个最近打开的项目以及New Project和Open Project按钮。要显示启动页面，选择Get Started选项。

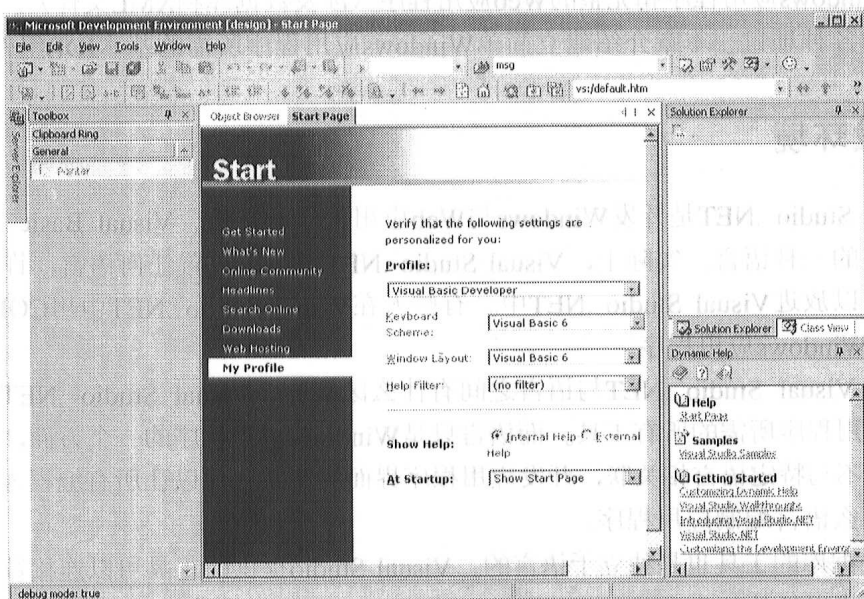


图1.1 首次运行Visual Studio时，出现这个窗口

其余选项使Downloads站点显示产品的更新信息，如新闻文章、更新文档和服务包或补丁。至少要经常切换到Downloads选项以检查更新内容。更新的安装应当是自动化的，在确认要下载和更新任何新组件之后自动进行。

Web Hosting选项打开一个页面，显示支持ASP.NET的ISP信息。需要通过这些ISP的服务将实际Web应用程序或Web服务发表到Internet中。Web应用程序与Web服务是Visual Studio中可以开发的两种项目（将在本书稍后介绍）。这些项目并不发布给用户，而是在Web服务器上运行。用户要连接到Web服务器的URL，并在浏览器中运行Web应用程序。

说明：产品的正式名称为Visual Studio .NET与Visual Basic .NET。本书称这个语言为VB.NET或VB，而称旧版语言为VB6。

启动新项目

现在就可以开始启动新项目和使用Visual Basic .NET了。为了介绍IDE的各种项目，我们准备建立一个简单窗体（还不是个应用程序）。窗体是应用程序的窗口，用户运行应用程序时，在桌面上看到这个窗体。

打开File菜单并选择New>Project对话框（如图1.2所示），出现Visual Studio中可以生成的项目类型清单。选择Windows Application模板，Visual Studio推荐项目名为WindowsApplication1，更名为MyTestApplication。项目名下面的另一个框是Location框，是生成新项目的文件夹（每个项目存放在不同文件夹中）。Visual Studio在Location框指定的文件夹中生成新文件夹，更名为项目名。可以保持默认项目文件夹，并单击OK按钮。

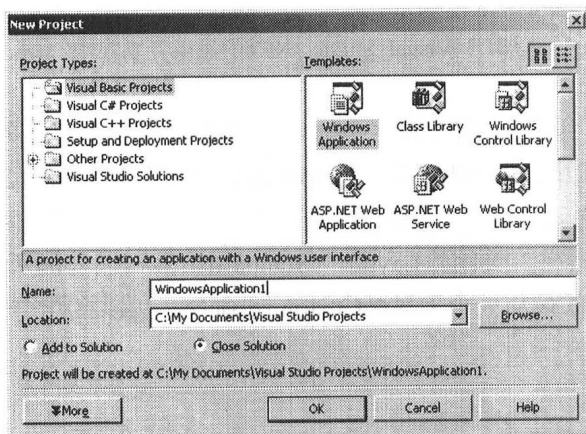


图1.2 New Project对话框

VB6→VB.NET

与旧版Visual Basic不同的是，Visual Basic .NET为项目建立新的文件夹并将项目文件保存在新的文件夹中，这是在编辑之前就完成的。IDE默认在每次运行项目时保存项目文件的改变。要改变这个行为，可以使用Tools>Options对话框，见本书稍后介绍。

这时Visual Studio IDE显示新项目的Form Designer（如图1.3所示）。主窗口是Form Designer，其中的灰色表面是设计模式中的新应用程序窗口。利用Form Designer，可以设计应用程序的显示界面（在窗体中放上Windows界面的各种组件），然后编制应用程序。

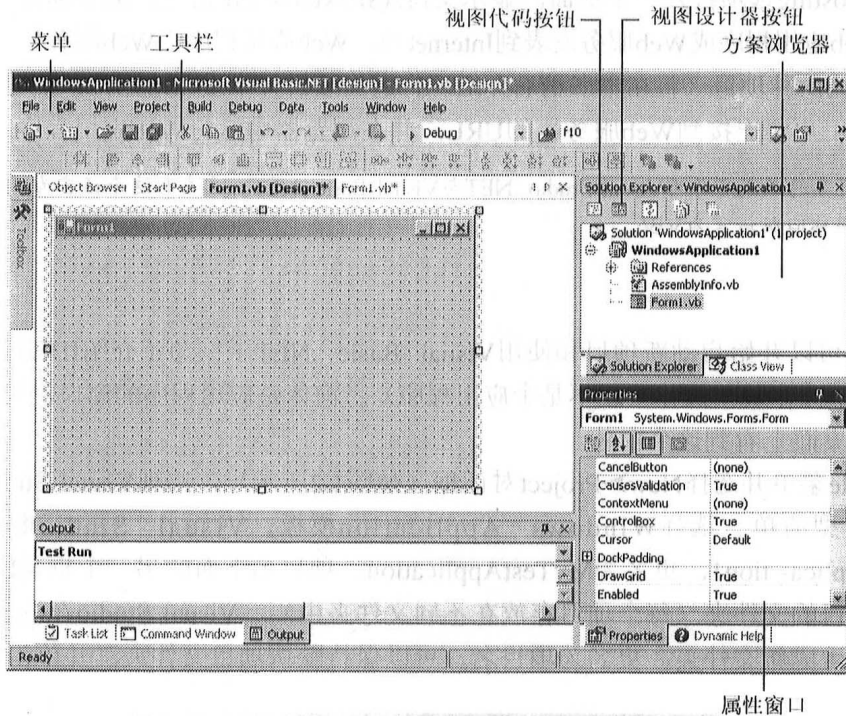


图1.3 Visual Studio .NET的集成开发环境

默认环境比较乱，因此下面隐藏几个开篇几章的项目中用不上的工具栏。可以随时显示任何工具栏。只要打开View菜单并选择Toolbars，就可以看到带有28个命令的子菜单，这些命令可在显示/隐藏之间切换。每个命令对应于一个工具栏，可以单击Toolbars子菜单中的命令显示/隐藏相应工具栏。关闭除Layout与Standard工具栏以外的所有其他工具栏。

Toolbars子菜单中的最后一个项目是Customize命令，可以打开一个对话框，指定要显示的命令和工具栏。

使用Windows Form Designer

要设计窗体，就要在窗体中放上运行时要向用户显示的所有控件。控件是Windows界面的组件（按钮、单选钮、清单，等等）。要打开工具箱，将指针移到最左边的Toolbox标签上，从而下拉Toolbox，如图1.4所示。这个工具箱包含窗体中可以使用的每个控件的图标。

控件组成标签，每个标签包含特定项目类型可用的控件。本书第一部分生成简单Windows应用程序，并使用Windows Forms标签中的控件。开发Web应用程序时，Windows Forms标签中的控件图标不可用，而只能在窗体中放上Web控件（是Web窗体，而不是本章建立的Windows窗体）。如果现在单击Web Forms标签，则其所有图标都关闭。

要将控件放到窗体中，可以双击控件图标，在窗体中放上具有默认长度的新实例。然后可以用鼠标移动和缩放。也可以用鼠标选择控件，然后在窗体中移动鼠标并绘制控件外形

线。控件新实例被放在窗体中，它填入用鼠标指定的矩形区。要将TextBox控件放在窗体中，双击工具箱中的TextBox图标。

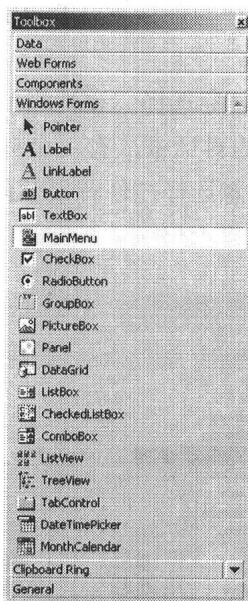


图1.4 Visual Studio .NET的Windows窗体工具箱

Properties窗口如图1.5所示，显示控件属性。这个窗口在IDE的最左端，显示窗体中所选控件的属性。如果看不到Properties窗口，可以选择View►Properties Window或按F4键。如果不选择控件，则显示Solution Explorer中所选项目的属性。将另一个TextBox控件放到窗体中。新控件放在上一控件上方。可以用鼠标在窗体中移动这两个控件的位置，然后右单击其中一个控件并从弹出菜单中选择Properties。

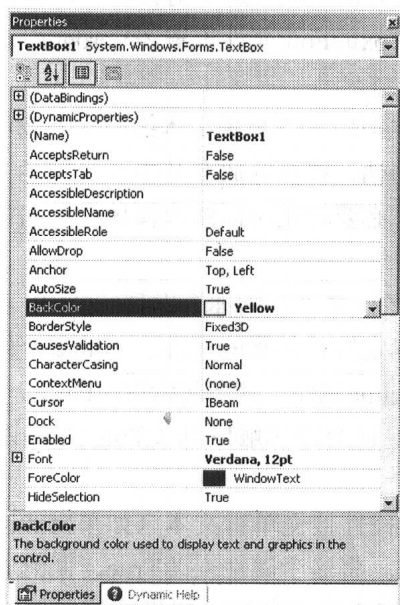


图1.5 TextBox控件属性

Properties窗口也称为Property Browser，有些属性确定控件外观，有些属性确定控件功能。找到TextBox控件的Text属性，并在属性名旁边的框中输入文本，将其设置为“My TextBox Control”（不含引号）。选择当前设置（TextBox1），并输入新字符串。控件Text属性为控件中显示的字符串。

然后找到其BackColor属性并用鼠标选择。属性当前设置旁边出现一个带箭头的按钮。单击这个按钮，可以看到图1.6所示的对话框，有三个标签（Custom、Web和System）。在这个对话框中，可以从三个标签中任意选择控件背景颜色。将控件背景颜色设置为黄色，注意窗体中控件外观的变化。

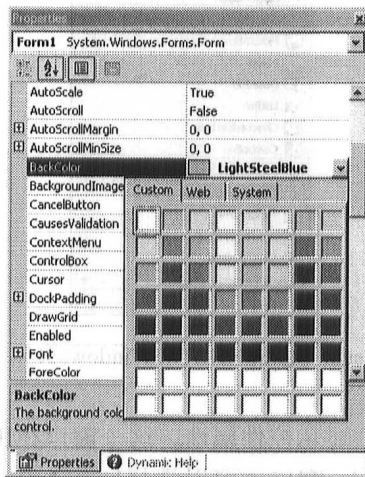


图1.6 在Properties对话框中设置控件背景颜色

然后找到控件的Font属性。可以单击属性名前面的加号，设置字体的各个属性，也可以单击带省略号的按钮，打开Font对话框。在这个对话框中，可以设置字体及其属性，然后单击OK按钮关闭对话框。将TextBox的Font属性设置为Verdana, 14磅，黑体。关闭Font对话框时，注意窗体中控件字体的变化。

也许控件Text属性指定的字符串不符合新字体绘制后的控件宽度。用鼠标选择窗体上的控件，可以看到其周围有八个句柄。将指针放在其中任一句柄上，它显示缩放控件的方向。让控件缩放成适合整个字符串，必要时还可以缩放窗体。单击窗体中某个地方，并在出现句柄时用鼠标缩放控件。

如果要使控件高度可容纳多行文本，则你会发现行不通。默认情况下，TextBox控件只能接受一行文本。前面操纵了确定控件外观的属性。下面要介绍的属性不仅确定控件外观，而且还确定控件功能。找到Multiline属性。其当前设置为False。展开设置清单并将其变为True（也可以双击属性名进行改变，这个操作触发开关True/False设置），然后切换到窗体中，选择TextBox控件并将其增高。

Multiline属性确定TextBox控件接受单行文本（如果Multiline = False）还是接受多行文本（如果Multiline = True）。将这个属性变为True，回到Text属性中，将其设置为长字符串并按Enter键。这时控件将文本分为多行。如果缩放控件，则行数改变，但整个字符串都放进控件中。这是因为控件WordWrap属性为True。将其设置为False看看字符串如何在控件上绘制。

多行Text Box控件通常有个垂直滚动栏，用户可以迅速找到所要的文本。找到控件的ScrollBars属性并单击带箭头的按钮展开设置清单。这个属性的设置为None、Vertical、Horizontal和Both。将其设置为垂直，将它的Text属性设置为长字符串，看看控件如何处理文本。设计时不能滚动控件上的文本。如果要移动滚动栏，则会移动整个控件。要在运行时检查控件行为，按F5键。这时会编译应用程序，过一会儿之后，桌面上出现带两个Text Box控件的窗体（如图1.7所示），这是应用程序用户所看到的（当然如果值得发布这个应用程序的话）。

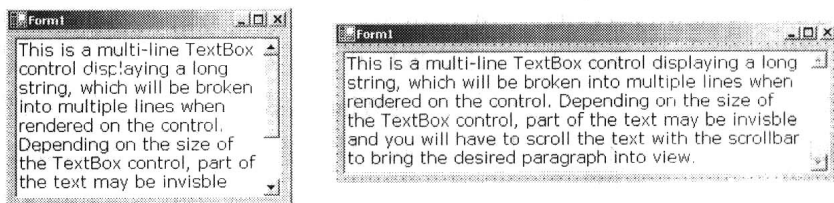


图1.7 显示多行文本的Text Box控件

在运行时，输入一些文本，选择控件上的文本，并按Ctrl+C键组将其复制到剪贴板中。也可以复制任何其他Windows应用程序中的文本并将其粘贴到Text Box控件中。完成之后，打开Debug菜单并选择Stop Debugging，从而终止应用程序执行，回到IDE中。

Text Box控件的CharacterCasing属性确定Text Box控件的功能，而不是确定Text Box控件的外观，其设置为Normal、Upper和Lower。在正常方式下，字符按输入时的样子显示。在Lower方式下，字符先自动转换成小写字母之后再在控件中显示。这个属性的默认设置为Normal。将其设置为Upper或Lower，再次运行应用程序，看看这个属性的值如何影响控件功能。在控件中输入一些小写字母，控件自动将其转换成大写；或者控件中输入一些大写字母，控件自动将其转换成小写。

设计新应用程序时，首先要设计应用程序窗体。窗体的设计确定应用程序的功能。效果上，窗体上的控件确定应用程序如何与用户交互。窗体本身是个原型，可以先向客户演示之后再实际编码。众所周知，将控件放到窗体中并设置属性之后，不需编制应用程序就可以实现许多功能。具有本节所介绍设置的Text Box控件是个可行的文本编辑器。

项目类型

下面先介绍除Windows应用程序之外，Visual Studio中还可以建立哪些项目类型。New Project对话框显示Visual Studio支持的所有项目类型，包括：

类库 (Class library) 类库是基本代码建立组件，没有显示界面，只是在项目中增加特定功能。简单地说，类是除当前项目之外的其他项目中使用的功能集合。但是，类不需要发布源代码。类库相当于VB6中的ActiveX DLL与ActiveX EXE项目类型。

Windows控件库 (Windows control library) 按钮与文本框之类的Windows控件（或简称控件）是用户界面的基本元素。如果Visual Basic所带的控件（默认在工具箱中出现）不能提供所需的功能，则可以建立定制控件。人们对特定操作设计定制控件，简化小组环境中大型应用程序的开发。如果你有很好的定制控件，就可以出售这个定制控件，计算机贸易杂志中有大量高级定制控件广告，补充现有的定制控件。

控制台应用程序 (Console application) 控制台应用程序具有一个非常有限的用户界面。这类应用程序在Command Prompt (命令提示) 窗口中显示输出, 从同一窗口接受用户输入。本章稍后将介绍简单控制台应用程序的例子, 这将是本书唯一的控制台应用程序。本书的目的是介绍如何建立具有丰富界面的Windows和Web应用程序, 而不是建立DOS风格应用程序。但是, 产品文档用控制台应用程序演示特定主题, 因此本章介绍一个简单的控制台应用程序。

Windows服务 (Windows service) Windows服务是NT服务的新名称, 是没有显示界面的长寿应用程序。这些服务可以在计算机打开、暂停和重新启动时自动启动。监视与响应文件系统改变的应用程序就可以实现为Windows服务。用户向特定文件夹上装文件时, Windows服务可以进行一些处理 (复制文件、读取内容和更新数据库, 等等)。本书不准备介绍Windows服务。

ASP.NET Web应用程序 (ASP.NET Web application) Web应用程序是Visual Studio最令人激动的新特性之一。Web应用程序就是放在Web服务器中, 满足通过浏览器进行的请求的应用程序。例如, 联机书店是个Web应用程序。Web服务器上运行的应用程序接受客户机请求 (客户机是带浏览器的远程计算机) 并以HTML页面形式返回请求响应。Web服务器并不新鲜, 但ASP.NET隐藏了建立Web应用程序的许多细节, 使这个过程与建立Windows应用程序非常相似。Web应用程序和Web服务将在本书后一部分详细介绍。

ASP.NET Web服务 (ASP.NET Web service) Web服务不等于Windows服务。Web服务程序放在Web服务器中, 像Web应用程序一样满足请求, 但不返回HTML页面, 而是返回计算结果或数据库查找。Web服务请求通常是由另一服务器进行的, 其负责处理数据。查询Sybex公司出版的所有VB图书的Web应用程序返回结果页面, 而接受相同查询的Web服务则返回结果的XML文件。请求应用程序用这个XML文件准备新页面, 并将其发送给客户机或填充Windows窗体。

Web控件库 (Web control library) 定制Windows控件可以在Windows窗体中使用, 而定制Web控件可以在Web页面中使用。本书不准备介绍Web控件, 但介绍ASP应用程序和Web应用程序如何与客户机交互之后, 就不难理解文档中的例子。

New Project对话框中的另外三个模板 (Empty Project、Empty Web Project和New Project In Existing Folder) 不是项目类型, 而只是组织新项目的方法。生成上述任何类型的项目时, Visual Studio按项目名生成一个新文件夹, 放进特定应用程序类型需要的文件。例如, Windows应用程序有窗体, 生成新Windows应用程序时, 项目文件夹中自动生成相应文件。对最后三种项目类型, 要自己生成和增加所有需要的项目。

第一个VB应用程序

本节开发一个简单应用程序, 不仅演示界面设计, 而且演示如何编制应用程序。我们建立的应用程序让用户输入喜欢的编程语言名称, 然后求值所作选择。VB比所有其他语言进一步, 得分为优, 而所有其他语言得分为良。

提示：本节建立的项目称为WindowsApplication1，在本书选配光盘第1章文件夹中。将本书选配光盘第1章文件夹中的WindowsApplication1文件夹复制到硬盘中，消除文件夹中的文件的只读属性。从光盘中复制的所有文件都是只读的，要改变这个属性（以便保存改变），可以选择项目文件夹中的所有文件，右单击并选择Properties。在出现的对话框中，清除Read-Only框。

可以打开本书选配光盘中的项目进行检查，但最好按照本段介绍的步骤建立项目。开始新项目，使用默认项目名WindowsApplication1，并在窗体中放上TextBox与Button控件。用鼠标在窗体中移动和缩放控件，如图1.8所示。

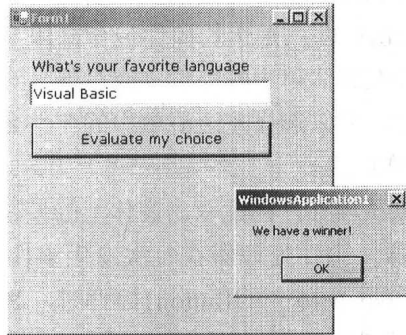


图1.8 处理用户输入字符串的简单应用程序

现在要插入一些代码，评估用户喜欢的语言。Windows应用程序是由小代码段构成的，称为事件处理器（event handlers），响应特定操作。本例中，我们要编程单击按钮的操作。用户单击按钮时，我们要执行一些代码，显示一个消息。

要插入Button控件的代码，需双击控件，可以看到应用程序的代码窗口，如图1.9所示，语句“Private ...”太长，无法放在一行中，因此我们用续行符（下划线）将其分成两行。语句太长时，可以用续行符将其分成多行。也可以打开编辑器的单词换行特性（稍后将介绍如何调整编辑器的属性）。注意我还在第一行代码的后半部分前面插入几个空格。习惯上，续行通过缩排与其他行分开。

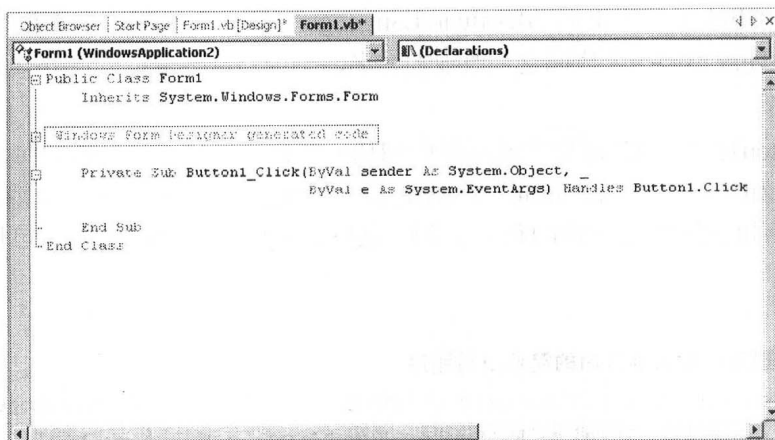


图1.9 处理Button控件Click事件的子程序

编辑器打开下列语句之间的子程序:

```
Private Sub Button1_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Button1.Click
End Sub
```

在Designer主窗格上方, 可以看到用窗体名命名的两个标签, 图1.9中为Form1.vb [Design]与Form1.vb标签。第一个标签是Windows Form Designer (在这里用图形化工具建立应用程序界面), 第二个标签是代码编辑器, 可以插入界面的代码。在代码编辑器顶端, 可以看到图1.9所示的两个ComboBoxes控件。左边的控件包含窗体上的控件名, 另一个控件包含每个控件可识别的事件名。选择左边清单中的控件(或对象)时, 另一个清单的内容相应调整。要编程特定控件的特定事件, 可以选择左边清单中的控件名(对象清单), 然后选择另一个清单中的事件名(事件清单)。

Click事件是Button控件的默认事件, 因此双击窗体上的按钮时, 进入Button1_Click子程序。这个子程序是个事件处理器。事件处理器在每次发生事件时自动调用。这里的事件是Button1控件的Click事件。每次单击窗体上的Button1控件时, 激活Button1_Click子程序。要响应Button1控件的Click事件, 就要在Button1_Click子程序中插入适当的代码。

子程序名为控件名下划线和事件名, 这只是默认名, 可以改变成任何名称(如EvaluateLanguage或StartCalculations)。语句末尾的关键字Handles使这个子程序成为事件处理器。关键字Handles告诉编译器这个子程序要处理什么事件。Button1.Click是Button1控件的Click事件。如果窗体中有另一个按钮Button2控件的Click事件, 则要编写处理Button2.Click事件的子程序代码。每个控件认识许多事件, 对每个控件与事件组合, 可以提供不同的事件处理器。当然, 我们并不需要对每个控件与事件组合编程事件处理器。

说明: 稍后将会介绍, 控件具有默认行为和能够处理基本事件。TextBox控件知道如何处理键击, CheckBox控件(带复选标志的小方框)通过每次单击时隐藏或显示复选标志而改变状态, ScrollBar控件在每次单击两端的箭头时移动指示符(控件中间的按钮)。由于控件具有默认行为, 因此窗体上的大多数控件事件不需要提供任何代码。

将Button1_Click子程序更名为EvaluateLanguage, 但不能改变这个子程序处理的事件名。如果在事件处理器中插入代码之后改变控件名, 则会自动改变子程序处理的事件名, 而不改变子程序名。

下面在Button1控件的Click事件处理器中增加一些代码。单击这个按钮时, 我们要检查控件中的文本, 如果是“Visual Basic”, 则显示一个消息, 否则显示另一个消息。在Private Sub与End Sub语句之间插入清单1.1所示语句(这里显示整个清单, 但第一句和最后一句不必重新输入)。

清单1.1 处理用户输入字符串的简单应用程序

```
Private Sub EvaluateLanguage_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Button1.Click
    Dim language As String
    language = TextBox1.Text
    If language = "Visual Basic" Then
```