



Beginning Visual Basic 2010

Visual Basic 2010

入门经典（第6版）

(美) Thearon Willis 著
Bryan Newsome
吴伟敏 李周芳 译



清华大学出版社

Visual Basic 2010 入门经典 (第 6 版)

(美) Thearon Willis 著
Bryan Newsome
吴伟敏 李周芳 译

清华大学出版社

北 京

Thearon Willis, Bryan Newsome

Beginning Visual Basic 2010

EISBN: 978-0-470-50222-8

Copyright © 2010 by Wiley Publishing, Inc.

All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2011-3823

本书封面贴有 Wiley 公司防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Visual Basic 2010 入门经典(第6版)/(美)威利斯(Willis, T.), (美)纽萨姆(Newsome, B.) 著; 吴伟敏, 李周芳 译.

—北京: 清华大学出版社, 2011.11

书名原文: Beginning Visual Basic 2010

ISBN 978-7-302-27005-8

I. ①V… II. ①威… ②纽… ③吴… ④李… III. ①BASIC 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 202603 号

责任编辑: 王 军 李维杰

装帧设计: 孔祥丰

责任校对: 成凤进

责任印制: 杨 艳

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 清华大学印刷厂

经 销: 全国新华书店

开 本: 185×260 印 张: 37 字 数: 993 千字

版 次: 2011 年 11 月第 1 版 印 次: 2011 年 11 月第 1 次印刷

印 数: 1~4000

定 价: 78.00 元

产品编号: 042930-01

作者简介

Thearon Willis 目前是一位高级开发人员，他使用 Visual Basic.NET 开发 Windows 应用程序和微软 Office 产品的插件。多年来，Thearon 从事过多种系统的开发工作，包括大型机系统和客户机-服务器系统。

Bryan Newsome 领导一个由顶尖开发人员组成的团队，致力于微软解决方案。自从开始建立 Visual Basic 5 解决方案以来，他掌握了 Visual Basic 的每个新版本，目前创建的所有新的解决方案都利用了 .NET 平台和 VB.NET。他为客户提供解决方案，并时刻关注着微软的最新技术。Bryan 是 Visual Basic.NET 方面的微软认证程序开发专家(MCPD)。

技术编辑简介

Damien Foggon 是一位开发人员、作家和先进技术的技术评论员，编辑了 50 余本有关.NET、C#、Visual Basic 和 ASP.NET 的图书。他是.NET 2.0 和.NET 3.5 的微软认证程序开发专家(MCPD)。联系方式是 <http://blog.littlepond.co.uk>。

致 谢

首先要感谢上帝，让我拥有可与他人共享的智慧和知识，感谢与我一起祈祷的许多人。还要感谢 Wiley 辛勤工作以使本书面世的所有人。如果我没有感谢我的好朋友、本书的合作者 Bryan Newsome，我就太过怠慢了，感谢你的辛勤工作和贡献。

——Thearon

感谢 Wiley 辛勤工作以使本书面世的所有人，特别感谢 Maureen Spears 帮助我按时完成通读检查工作。当然还要感谢 Thearon Willis 对本书所做的再次校对。

——Bryan

前言

Visual Basic 2010 是微软公司较为流行的 VB.NET 编程语言的最新版本，是 Visual Studio 2010 环境支持的几种语言之一。Visual Basic 2010 的优势在于其易用性和高速度，利用该语言可以轻松、快捷地编写出 Windows Forms 应用程序、WPF Windows 应用程序、Web 应用程序、WPF Browser 应用程序、移动设备应用程序和 Web 服务。

本书将全面介绍 Visual Basic 2010 的各种编程知识，展示如何创建上述应用程序和服务。同时，您也可以学习面向对象的编程技术，掌握创建业务对象和 Windows 控件的方法。

微软的 .NET Framework 使 Visual Basic 2010 程序员能像 C# 或 C++ 程序员一样创建完全面向对象的各種程序。 .NET Framework 提供了一套通用的基类，Visual Studio 2010 中的所有编程语言都可以使用它们，这样一来，程序员在创建面向对象的程序方面就拥有与 C# 或 C++ 一样的功能了。

本书将使您在 Visual Basic 2010 编程方面打下坚实的基础，并给您带来编程的无限乐趣。

0.1 本书读者对象

本书的目的是教您如何尽可能快捷地运用 Visual Basic 2010 编写有用的程序。

本书适合于以下两类初学者：

- 一类是编程的初学者，他们选择从 Visual Basic 2010 开始学起。这是个很好的选择！Visual Basic 2010 不但易学易用，而且功能强大。
- 另一类是曾用其他语言进行编程，但初次接触 .NET 编程知识的人员。这类初学者现在选择学习 Visual Basic 2010 是一个不错的主意。不管您熟悉的语言是 Fortran 还是 VB 6，本书都将帮助您迅速掌握 Visual Basic 2010。

0.2 本书主要内容

Visual Basic 2010 提供了工具和语言方面的大量功能。当然，一本书不可能包含 Visual Basic 2010 的所有内容，这些内容要用一个书库来容纳。本书的目的是让您尽快入门，它相当于一个路标，可以指导您的学习。在介绍创建应用程序的基础知识(如创建窗口和控件，如何用代码处理突发事件，什么是面向对象的程序设计以及如何在应用程序中使用它，等等)之余，本书还为您提供了亲自动手的机会。本书的组织结构如下：

- 第 1~9 章介绍 Visual Studio 2010 和 Windows 编程的基础知识。
- 第 6 章介绍 XAML 和 WPF(Windows Presentation Foundation)编程。
- 第 10 章阐述应用程序的调试和错误处理方法。
- 第 11~13 章介绍面向对象编程技术以及如何创建对象。

- 第 14 章介绍如何创建 Windows Forms 用户控件。
- 第 15、16 章介绍如何利用数据库进行编程,同时涵盖 Access、SQL Server 及 ADO.NET 等知识。
- 第 17、18 章简述动态数据网站、ASP.NET,并讨论如何编写 Web 应用程序。
- 第 19 章对 XML 进行简要介绍。XML 这种功能强大的工具能把各种应用程序结合起来,而无需考虑所使用的语言。
- 第 20 章介绍使用 ClickOnce(一键部署)技术部署应用程序。

0.3 Visual Basic 2010 所需的运行环境

除了要有学习的热忱之外,学习前 15 章还需要一台能够运行 Windows 7(首选)、Windows Vista、Windows XP(家庭版或专业版)、Windows Server 2008 或 Windows Server 2003 系统,并且安装了 Internet Explorer 的计算机。当然,还要安装下列软件之一:

- Microsoft Visual Basic 2010 Professional Edition
- Microsoft Visual Basic 2010 Premium Edition
- Microsoft Visual Basic 2010 Ultimate Edition
- Microsoft Visual Basic 2010 Team System

0.4 源代码

在练习书中的示例时,可以选择手动输入代码或者使用本书附带的源代码文件。书中用到的所有源代码都可以从 www.wrox.com 下载。进入站点 <http://www.worx.com> 后,只需找到本书的书名(使用 Search 搜索框或书名列表),单击本书详细信息页面上的 Download Code 链接,就可以得到本书所有的源代码。



注意:因为很多书的书名都相似,所以用 ISBN 搜索更为容易。本书英文版的 ISBN 是 978-0-470-50222-8。

下载完代码后,用您喜欢的压缩工具把它解压缩。此外,也可以去 Wrox 的主下载页面 www.worx.com/dynamic/books/download.aspx 找到本书或 Wrox 出版的其他书籍的代码。

0.5 勘误表

尽管我们竭尽所能来确保在正文和代码中没有错误,但人无完人,错误难免会发生。如果您在 Wrox 出版的书中发现了错误(例如拼写错误或代码错误),我们将非常感谢您的反馈。发送勘误表将节省其他读者的时间,同时也会帮助我们提供更高质量的信息。

要找到本书的勘误页面，可以进入 www.worx.com，使用 Search 搜索框或书名列表定位本书，然后在本书的详细信息页面上单击 Book Errata 链接。在这个页面上可以查看为本书提交的、Wrox 编辑粘贴上去的所有错误。完整的书名列表(包括每本书的勘误表)也可以从 www.wrox.com/misc-pages/booklist.shtml 上获得。

如果您在本书的勘误页面上没有看到您发现的错误，可以到 www.wrox.com/contact/techsupport.shtml 上填写表单，把您发现的错误发给我们。我们会检查这些信息，如果属实，就把它添加到本书的勘误页面上，并在本书随后的版本中更正错误。

0.6 p2p.wrox.com

如果想和作者或同行进行讨论，请加入 <http://p2p.worx.com> 上的 P2P 论坛。该论坛是一个基于 Web 的系统，您可以发布有关 Wrox 图书及相关技术的消息，与其他读者或技术人员交流。该论坛提供了订阅功能，当您感兴趣的主题有新帖子发布时，系统会邮件通知。Wrox 的作者、编辑、其他业界专家和像您一样的读者都会出现在这些论坛中。

在 <http://p2p.wrox.com> 网站上，您会找到很多不同的论坛，它们不但有助于您阅读本书，还有助于您开发自己的应用程序。加入论坛的步骤如下：

- (1) 进入 <http://p2p.wrox.com>，单击 Register 链接。
- (2) 阅读使用条款，然后单击 Agree 按钮。
- (3) 填写加入该论坛必需的信息和其他您愿意提供的信息，单击 Submit 按钮。
- (4) 您将收到一封电子邮件，描述如何验证您的账户和完成加入过程。



注意：不加入 P2P 也可以阅读论坛里的消息。但是如果要发布自己的消息，就必须加入。

加入之后，就可以发布新的消息和回复其他用户发布的消息。可以随时在 Web 上阅读论坛里的消息。如果想让某个论坛的新消息以电子邮件的方式发给您，可以单点击论坛列表中论坛名称旁边的 Subscribe to this Forum 图标。

要了解如何使用 Wrox P2P 的更多信息，请阅读 P2P FAQ，其中回答了论坛软件如何使用的问题，以及许多与 P2P 和 Wrox 图书相关的问题。要阅读 FAQ，单击任何 P2P 页面上的 FAQ 链接即可。

目 录

第 1 章 初识 Visual Basic 2010	1
1.1 事件驱动编程	2
1.2 安装 Visual Basic 2010	3
1.3 Visual Basic 2010 集成开发环境	5
1.3.1 初始化开发环境	6
1.3.2 菜单	6
1.3.3 工具栏	7
1.4 创建简单的应用程序	8
1.4.1 工具箱	11
1.4.2 改进型匈牙利表示法	13
1.4.3 代码编辑器	14
1.5 使用帮助系统	17
1.6 小结	18
1.7 练习	18
第 2 章 .NET Framework 概述	19
2.1 微软的信心来自于 Windows	19
2.1.1 MSN 1.0	20
2.1.2 .NET 理念	20
2.1.3 .NET 与 Java 有些类似	21
2.1.4 现在所处的阶段	22
2.2 为 Windows 编写软件	22
2.2.1 .NET Framework 类	23
2.2.2 执行代码	24
2.3 公共语言运行时	25
2.3.1 载入并执行代码	25
2.3.2 应用程序隔离	25
2.3.3 安全性	25
2.3.4 互操作	26
2.3.5 异常处理	26
2.4 通用类型系统和公共语言规范	26

2.5 小结	27
第 3 章 编写软件	29
3.1 信息和数据	29
3.1.1 算法	29
3.1.2 编程语言	30
3.2 变量	31
3.3 注释和空白	33
3.3.1 注释	33
3.3.2 空白	35
3.4 数据类型	35
3.4.1 数字类型	35
3.4.2 常见的整型数学运算	36
3.4.3 整型运算的缩写方式	38
3.4.4 浮点型数学运算	40
3.4.5 字符串类型	42
3.4.6 使用日期	49
3.4.7 布尔类型	54
3.5 存储变量	55
3.5.1 二进制	55
3.5.2 位和字节	55
3.5.3 值的表示	56
3.5.4 值的转换	57
3.6 方法	58
3.6.1 使用方法的原因	59
3.6.2 前面曾介绍过的方法	59
3.6.3 创建方法	62
3.6.4 选择方法名称	65
3.6.5 作用域	65
3.7 小结	67
3.8 练习	68
第 4 章 流程控制	69
4.1 作出决策	69

4.2	If 语句	70	5.4.2	为结构添加属性	126
4.2.1	Else 语句	71	5.5	数组列表	127
4.2.2	可进行多重选择的 ElseIf 语句	72	5.5.1	使用数组列表	127
4.2.3	嵌套的 If 语句	73	5.5.2	在数组列表中进行删除操作	131
4.2.4	单行 If 语句	73	5.5.3	显示数组列表中的项	134
4.2.5	比较运算符	74	5.6	集合	134
4.2.6	字符串比较	81	5.6.1	创建 CustomerCollection 类	135
4.3	Select Case 语句	83	5.6.2	添加 Item 属性	136
4.3.1	不区分大小写的 Select Case 语句	86	5.7	使用散列表创建查询表	138
4.3.2	多重选择	88	5.7.1	使用散列表	138
4.3.3	Case Else 语句	89	5.7.2	Remove、RemoveAt 和 Clear 方法	141
4.3.4	Select Case 中的不同数据类型	89	5.7.3	区分大小写	143
4.4	循环	90	5.8	高级数组操作	144
4.4.1	For...Next 循环	90	5.8.1	动态数组	144
4.4.2	Do...Loop 循环	95	5.8.2	使用 Preserve 关键字	146
4.4.3	嵌套循环	99	5.9	小结	147
4.4.4	提前退出循环	100	5.10	练习	147
4.4.5	死循环	102			
4.5	小结	103	第 6 章	可扩展应用程序标记语言 (XAML)	149
4.6	练习	104	6.1	什么是 XAML	149
第 5 章	处理数据结构	105	6.2	XAML 语法	150
5.1	数组	105	6.3	WPF	153
5.1.1	定义和使用数组	105	6.3.1	创建一个富 WPF 用户界面	154
5.1.2	使用 For Each...Next 循环	108	6.3.2	使用 WPF 通用控件	159
5.1.3	将数组作为参数传递	110	6.3.3	连接事件	163
5.1.4	数组的排序	112	6.4	小结	166
5.1.5	反向检索	113	6.5	练习	167
5.1.6	用值初始化数组	115	第 7 章	构建 Windows 应用程序	169
5.2	枚举	116	7.1	响应事件	169
5.2.1	使用枚举	116	7.2	创建简单的应用程序	176
5.2.2	确定状态	119	7.2.1	构建窗体	176
5.2.3	设置无效值	121	7.2.2	统计字符	179
5.3	常量	121	7.2.3	统计单词	181
5.3.1	使用常量	122	7.3	创建更复杂的应用程序	187
5.3.2	各种常量类型	123	7.3.1	Text Editor 项目	187
5.4	结构	123	7.3.2	创建工具栏	187
5.4.1	创建结构	124			

7.3.3 创建状态栏	191	8.7.1 FolderBrowserDialog 控件 的属性	238
7.3.4 创建编辑框	193	8.7.2 使用 FolderBrowserDialog 控件	238
7.3.5 清除编辑框	194	8.8 小结	241
7.3.6 响应工具栏按钮	197	8.9 练习	242
7.4 使用多个窗体	202	第 9 章 创建菜单	243
7.5 小结	205	9.1 菜单功能概述	243
7.6 练习	206	9.1.1 图形	243
第 8 章 显示对话框	207	9.1.2 访问键	244
8.1 MessageBox 对话框	207	9.1.3 快捷键	244
8.1.1 MessageBox 中的可用 图标	208	9.1.4 复选标记	244
8.1.2 MessageBox 中的可用 按钮	208	9.1.5 Properties 窗格	245
8.1.3 设置默认按钮	209	9.2 创建菜单	246
8.1.4 其他选项	209	9.2.1 设计菜单	246
8.1.5 Show 方法的语法	209	9.2.2 添加工具栏和控件	247
8.1.6 MessageBox 的示例	211	9.2.3 为菜单编写代码	249
8.2 OpenFileDialog 控件	214	9.2.4 为 View 菜单和工具栏编写 代码	253
8.2.1 OpenFileDialog 控件	214	9.2.5 测试代码	254
8.2.2 OpenFileDialog 的属性	215	9.3 上下文菜单	256
8.2.3 OpenFileDialog 的方法	216	9.3.1 创建上下文菜单	256
8.2.4 使用 OpenFileDialog 控件	216	9.3.2 激活或禁用菜单项和工具栏 按钮	259
8.3 SaveFileDialog 控件	220	9.4 小结	262
8.3.1 SaveFileDialog 的属性	221	9.5 练习	263
8.3.2 SaveFileDialog 的方法	221	第 10 章 调试和错误处理	265
8.3.3 使用 SaveFileDialog 控件	221	10.1 主要错误类型	265
8.4 FontDialog 控件	225	10.1.1 语法错误	266
8.4.1 FontDialog 的属性	225	10.1.2 执行错误	269
8.4.2 FontDialog 的方法	225	10.1.3 逻辑错误	269
8.4.3 使用 FontDialog 控件	225	10.2 调试	270
8.5 ColorDialog 控件	228	10.2.1 创建示例项目	270
8.5.1 ColorDialog 的属性	228	10.2.2 设置断点	285
8.5.2 使用 ColorDialog 控件	229	10.2.3 使用 Watch 窗格和 QuickWatch 对话框 进行调试	291
8.6 PrintDialog 控件	230	10.2.4 使用 Autos 窗格进行 调试	293
8.6.1 PrintDialog 的属性	231		
8.6.2 使用 PrintDialog 控件	231		
8.6.3 PrintDocument 类	231		
8.6.4 打印文档	232		
8.7 FolderBrowserDialog 控件	238		

10.2.5 使用 Locals 窗格进行调试	293	第 12 章 面向对象的高级技术	333
10.3 错误处理	294	12.1 构建收藏夹浏览应用程序	333
10.4 小结	298	12.1.1 Internet 快捷方式和收藏夹	333
10.5 练习	298	12.1.2 使用类	336
第 11 章 构建对象	299	12.1.3 遍历收藏夹	341
11.1 对象的概念	299	12.1.4 查看收藏夹	347
11.1.1 封装	300	12.2 另一个收藏夹浏览应用程序	349
11.1.2 方法和属性	300	12.2.1 构建收藏夹的系统托盘图标	350
11.1.3 事件	301	12.2.2 显示收藏夹	351
11.1.4 可见性	301	12.3 使用共享属性和方法	354
11.1.5 类的概念	302	12.3.1 使用共享属性	355
11.2 构建类	302	12.3.2 使用共享方法	358
11.3 重用性	303	12.4 理解面向对象编程和内存管理	360
11.4 设计对象	304	12.4.1 垃圾回收	361
11.4.1 状态	305	12.4.2 释放资源	361
11.4.2 行为	305	12.4.3 整理碎片和压缩	362
11.4.3 存储状态	305	12.5 小结	363
11.4.4 属性	308	12.6 练习	364
11.4.5 读/写属性	311	第 13 章 建立类库	365
11.4.6 IsMoving 方法	313	13.1 理解类库	365
11.5 构造函数	315	13.1.1 创建类库	366
11.6 继承	317	13.1.2 为 Favorites Viewer 项目创建类库	367
11.6.1 添加新方法和属性	318	13.1.3 多层应用程序	370
11.6.2 添加 GetPowerToWeightRatio 方法	320	13.2 使用强名	371
11.6.3 改变默认值	321	13.2.1 对程序集签名	371
11.6.4 多态性	323	13.2.2 程序集的版本	373
11.6.5 重写更多的方法	324	13.3 注册程序集	374
11.6.6 从对象类中继承	326	13.3.1 Gacutil 实用程序	374
11.7 对象和结构	326	13.3.2 程序集在 References 对话框中不可见的原因	374
11.8 .NET Framework 类	326	13.4 设计类库	375
11.8.1 名称空间	327	13.5 使用第三方类库	375
11.8.2 Imports 语句	328	13.6 用对象浏览器查看类	376
11.8.3 创建自己的名称空间	329	13.7 小结	377
11.8.4 .NET Framework 中的继承	331	13.8 练习	378
11.9 小结	332		
11.10 练习	332		

第 14 章 创建 Windows 窗体用户控件379	
14.1 Windows 窗体控件.....379	
14.2 创建和测试用户控件.....380	
14.3 给用户控件添加属性、方法和事件.....383	
14.3.1 给用户控件添加属性.....383	
14.3.2 给用户控件添加方法.....385	
14.3.3 给用户控件添加事件.....386	
14.4 设计期间或运行期间.....389	
14.5 创建 Command Link 控件.....391	
14.5.1 创建 Command Link 控件.....392	
14.5.2 使用 Command Link 控件.....399	
14.6 小结.....402	
14.7 练习.....402	
第 15 章 访问数据库405	
15.1 数据库的概念.....405	
15.1.1 Microsoft Access 对象.....406	
15.1.2 表.....406	
15.1.3 查询.....406	
15.2 SQL 中的 SELECT 语句.....407	
15.3 Access 中的查询.....408	
15.4 数据访问组件和控件.....412	
15.4.1 DataSet 组件.....412	
15.4.2 DataGridView 控件.....413	
15.4.3 BindingSource 组件.....413	
15.4.4 BindingNavigator 控件.....413	
15.4.5 TableAdapter 组件.....413	
15.5 数据绑定.....414	
15.6 小结.....419	
15.7 练习.....420	
第 16 章 使用 SQL Server 和 ADO.NET 进行数据库编程421	
16.1 ADO.NET.....422	
16.1.1 ADO.NET 数据名称空间.....422	
16.1.2 SqlConnection 类.....423	
16.1.3 SqlCommand 类.....425	
16.1.4 SqlDataAdapter 类.....427	
16.1.5 DataSet 类.....431	
16.1.6 DataView 类.....431	
16.2 使用 ADO.NET 类.....433	
16.3 数据绑定.....441	
16.3.1 BindingContext 和 CurrencyManager 对象.....441	
16.3.2 绑定控件.....442	
16.4 小结.....468	
16.5 练习.....468	
第 17 章 动态数据网站469	
17.1 创建一个动态数据 LINQ to SQL 网站.....469	
17.2 定制动态数据网站的设计.....473	
17.3 小结.....478	
17.4 练习.....478	
第 18 章 ASP.NET479	
18.1 瘦客户端体系结构.....479	
18.2 Web 窗体和 Windows 窗体的比较.....480	
18.2.1 Windows 窗体的优点.....480	
18.2.2 Web 窗体的优点.....481	
18.3 Web 应用程序基础.....481	
18.3.1 Web 服务器.....482	
18.3.2 浏览器.....482	
18.3.3 HTML.....482	
18.3.4 JavaScript.....482	
18.3.5 CSS.....482	
18.4 关于 ASP 页面.....483	
18.4.1 ASP.NET Web 页面的优点.....483	
18.4.2 特殊的 Web 站点文件.....483	
18.4.3 开发方式.....483	
18.4.4 工具箱.....484	
18.5 构建 Web 应用程序.....484	
18.5.1 为客户端和服务端处理创建 Web 窗体.....484	

18.5.2	Visual Studio 2010 中 Web 站点的位置	489	19.3	集成地址簿应用程序	533
18.5.3	数据的输入和验证	491	19.3.1	集成规则	533
18.5.4	设计布局、主题和导航	496	19.3.2	从其他应用程序中读取 地址簿	534
18.5.5	使用 GridView 构建数据 驱动的 Web 窗体	501	19.4	小结	539
18.6	小结	505	19.5	练习	540
18.7	练习	506	第 20 章	部署应用程序	541
第 19 章	Visual Basic 2010 和 XML	507	20.1	部署简介	541
19.1	了解 XML	507	20.1.1	ClickOnce 部署	542
19.1.1	XML 简介	508	20.1.2	XCOPY 部署	546
19.1.2	向 Visual Basic 编程新手 介绍 XML	510	20.2	创建 Visual Studio 2010 安装 程序	547
19.1.3	规则	510	20.3	UI 编辑器	550
19.2	地址簿项目	510	20.4	部署不同的解决方案	553
19.2.1	创建项目	511	20.4.1	私有程序集	553
19.2.2	SerializableData 类	512	20.4.2	共享程序集	554
19.2.3	载入 XML 文件	517	20.4.3	部署桌面应用程序	554
19.2.4	修改数据	519	20.4.4	部署 Web 应用程序	554
19.2.5	发送 email	520	20.4.5	部署 XML Web 服务	555
19.2.6	创建地址列表	521	20.4.6	一些有用的工具	555
19.2.7	忽略成员	525	20.5	小结	555
19.2.8	载入地址	527	20.6	练习	556
19.2.9	添加新地址	528	附录 A	练习题答案	557
19.2.10	地址导航	529	附录 B	如何深入学习	569
19.2.11	删除地址	531			

第 1 章

初识 Visual Basic 2010

本章主要内容:

- 事件驱动编程
- Visual Basic 2010 的安装
- Visual Basic 2010 集成开发环境(Integrated Development Environment, IDE)
- 创建简单的 Windows 程序
- 使用集成帮助系统

用 Visual Basic 2010 和 Windows 7 进行编程是令人兴奋的。Windows 7 是微软最新发布的 Windows 操作系统,其中包含许多新功能,使得 Windows 编程非常有趣。Windows 用户界面和 Visual Basic 2010 中的重大改变,非常便于编写专业的 Windows 应用程序、Web 应用程序和 Web 服务。还没有升级到 Windows 7? 不必担心, Visual Basic 2010 也可用于 Windows 的以前版本编写专业的应用程序。

即使您以前从未编写过任何程序,通过本书的学习,也仍然可以快速掌握 Visual Basic 2010 编程语言。我们将循序渐进地介绍 Visual Basic 2010 的基础知识,使您相信自己可以快速掌握它。

编写计算机程序非常类似于教小孩系鞋带。要想成功,必须找到给出指令的正确方式。使用 Visual Basic 2010 语言就可以告诉计算机该如何去做。但计算机像小孩子一样,只有把事情解释得非常清楚,它才会明白。如果您以前从未编写过程序,可能学习起来会费劲些。然而, Visual Basic 2010 是一种易用的、但能解释某些复杂事件的语言。能理解最基础的知识固然很好,而 Visual Basic 2010 能把编程人员从编写复杂 Windows 程序的繁琐工作中解放出来,让他们集中精力去解决实际问题。

Visual Basic 2010 有助于创建运行在 Windows 操作系统(如 Windows 7、Windows Server 2008 和 Windows Mobile 6.1)下的应用程序。如果正在看这本书,那么您可能希望或需要创建这类程序。即使您以前从未接触过编程,但通过阅读本书中“试一试”部分的内容,也会逐渐熟悉 Visual Basic 2010 语言的方方面面,以及 Microsoft .NET Framework 的基础知识。您会发现,其实它比我们想象的要容易。在了解这一点之前,您将感受到可以轻松地使用 Visual Basic 2010 创建各种不同的程序。

Visual Basic 2010 也可用于创建 Web 应用程序和 Web 服务,以及可在 Pocket PC 或智能手机上

运行的移动应用程序。然而，在扩展到其他平台之前，您应该先学习 Windows 应用程序。

1.1 事件驱动编程

Windows 程序完全不同于过去的 MS-DOS 程序。DOS 程序自始至终遵循一种相对严格的路径。虽然这不一定限制程序的功能，但却限制了用户编程时采用的方法。编写 DOS 程序就像穿过一条走廊，要到达终点，必须穿过整条走廊，清除途中可能遇到的所有障碍，而沿途只能打开一些特定的门。

另一方面，Windows 提供了事件驱动编程的方法。这些事件包括：单击按钮、改变窗口大小、改变文本框中的项等。我们编写的代码可以响应这些事件。用走廊的示例作类比就是：在 Windows 程序中，要到达走廊的终点，只需单击一下终点就行了，而不用考虑这个走廊。如果到达了终点，却发现这不是我们想去的地方，那么可以直接奔向新的目的地，而不用返回起点。程序会对这个动作作出反应，并且执行必要的动作来完成指定的任务。

Windows 编程的另一大优点是硬件的抽象化。这就是说，Windows 负责与硬件的通信。我们不需要知道市场上每种激光打印机的内部工作原理，就可以输出结果；也不需要研究图形卡的原理就可以编写游戏。Windows 提供了能与硬件制造商编写的驱动程序通信的通用例程，并封装了这个功能。这就是为什么 Windows 如此成功的主要原因。这些通用例程称作 Windows API(应用编程接口)，.NET Framework 中的类负责与这些 API 通信。

在 1991 年 Visual Basic 1.0 问世以前，开发人员必须精通 C 和 C++ 编程以及 Windows 系统本身的基本构建块(即 Windows API)。这一复杂性意味着，只有经过正规培训的专业人员才能编写出在 Windows 上运行的软件。Visual Basic 的问世改变了这一切，据估计，现在用 Visual Basic 语言编写的代码量远高于使用其他任何一种语言编写的代码量。

Visual Basic 取消了用户界面(UI)繁琐的代码编写工作，使 Windows 编程焕然一新。它允许程序员绘制自己的 UI，以便集中精力去解决一些业务问题。一旦创建了 UI，程序员就可向其中添加代码来响应事件。

Visual Basic 从一开始就是可扩展的。第三方供应商很快看到了对开发人员有帮助的可复用模块的市场。这些模块或控件原来称作 VBX(以其文件扩展名命名)。在 Visual Basic 5.0 之前，如果您不喜欢某一按钮的使用方式，那么可以购买或自定义该按钮。但是，这些控件必须用 C 或 C++ 编写。数据库访问实用程序就是这类控件中最早的一批。Visual Basic 5 版本中引入了 ActiveX 的概念，允许开发人员创建自己的 ActiveX 控件。

Visual Basic 3.0 的出现又一次改变了编程世界。现在可以用 Visual Basic 来创建可供用户直接访问的数据库应用程序(即所谓的前端应用程序)，不再需要依赖第三方的控件。微软是使用数据访问对象(DAO)来完成这项任务的，它允许程序员像操作用户界面那样来处理数据。

为了把开发人员吸引到 Windows 95 平台，Visual Basic 3.0 又升级为 Visual Basic 4.0 和 5.0。它们使开发人员更容易编写代码，而这些代码又可供使用其他语言的开发人员使用。Visual Basic 6.0 集成了 ActiveX Data Objects(ADO)，提供了一种访问数据库的全新方法。微软开发 ADO，是为了帮助 Web 开发人员使用 Active Server Pages(ASP)来访问数据库。经过多年的改进后，Visual Basic 已经在编程领域占据了主导地位。它可以帮助开发人员在数字化时代编写出健壮的、可维护的应用程序。