

PROGRAMMER TO PROGRAMMER™



Beginning Microsoft Visual Basic 2008

Visual Basic 2008

入门经典

(第5版)

(美) Thearon Willis
Bryan Newsome
徐燕华

著
译



清华大学出版社



Beginning Microsoft Visual Basic 2008

Visual Basic 2008

入门经典

(第5版)

作者: Thomas I. Powell 著
译者: 陈旭 等
ISBN: 978-7-304-05411-1



中国水利水电出版社

Visual Basic 2008 入门经典 (第 5 版)

(美) Thearon Willis 著
Bryan Newsome
徐燕华 译

清华大学出版社

北 京

Thearon Willis, Bryan Newsome
Beginning Microsoft Visual Basic 2008
EISBN: 978-0-470-19134-7

Copyright © 2008 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: 01-2009-2131

本书封面贴有 John Wiley & Sons 公司防伪标签, 无标签者不得销售。
版权所有, 翻印必究。举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Visual Basic 2008 入门经典(第5版)/(美) 威利斯(Willis, T.), (美) 纽萨姆(Newsome, B.) 著; 徐燕华 译.
—北京: 清华大学出版社, 2009.4

书名原文: Beginning Microsoft Visual Basic 2008

ISBN 978-7-302-19473-6

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

中国版本图书馆 CIP 数据核字(2009)第 016295 号

责任编辑: 王 军 于 平

装帧设计: 孔祥丰

责任校对: 成凤进

责任印制: 何 芊

出版发行: 清华大学出版社

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

社 总 机: 010-62770175

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

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

邮 购: 010-62786544

印 刷 者: 清华大学印刷厂

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 49 字 数: 1192 千字

版 次: 2009 年 4 月第 1 版 印 次: 2009 年 4 月第 1 次印刷

印 数: 1~4000

定 价: 98.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 029294-01

前言

Visual Basic 2008 是微软公司较为流行的 VB.NET 编程语言的最新版本,是 Visual Studio 2008 环境中所支持的几种语言之一。Visual Basic 2008 的优势在于其易用性和速度,利用该语言可以轻松而快捷地编写出 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 2008 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 2008 程序员像 C#或 C++程序员一样能够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 2008 中的所有编程语言都可以使用它们,这样对于程序员来说,使用 Visual Basic 2008 就像使用 C#或 C++一样可以创建面向对象的程序。

本书将使您在 Visual Basic 2008 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 2008 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 2008 开始学起。这是个很好的选择! Visual Basic 2008 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 编程知识的初学者。这类初学者现在选择学习 Visual Basic 2008 是一个不错的主意。不管您熟悉的语言是 Fortran 还是 VB6,本书都将帮助您迅速掌握 Visual Basic 2008。

0.2 本书主要内容

Visual Basic 2008 提供了大量的工具和语言方面的功能。当然,不可能用一本书包含 Visual Basic 2008 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)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 章介绍了 Visual Studio 2008 和 Windows 编程的基础知识。
- 第 6 章介绍了 XAML 和 Windows Presentation Foundation(WPF)编程。
- 第 10 章阐述了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 章介绍了如何创建 Windows Forms 用户控件。
- 第 15 章介绍 Windows 应用程序中的图形知识。
- 第 16、17 章介绍如何利用数据库进行编程,同时涵盖 Access、SQL Server、ADO.NET 及 LINQ 等知识。
- 第 18、19 章简述了 ASP.NET 并讨论如何编写 Web 应用程序。
- 第 20 章对 XML 作了简要的介绍。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具体语言。
- 第 21 章介绍了 Web 服务(Web Services)和 WCF(Windows Communication Foundation)。
- 第 22 章介绍使用 Windows Workflow Foundation(WF)创建顺序工作流。
- 第 23 章讨论如何使用 Compact Framework 类创建移动设备程序。
- 第 24 章介绍使用 ClickOnce 技术部署应用程序。
- 第 25 章介绍了如何深入学习 Visual Basic 2008。
- 附录 A 给出了各章练习题的答案。
- 附录 B 介绍了 Microsoft Solution Framework。
- 附录 C 讲述了关于安全的背景知识。
- 附录 D 介绍了 Windows CardSpace。
- 附录 E 比较了最新版本的 .NET Framework 之间的不同点。

0.3 Visual Basic 2008 所需的运行环境

除了要有一番学习的热忱之外,学习前 13 章还需要一台能够运行 Windows Vista(首选)、Windows XP(家庭版或专业版)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 2008 Express Edition
- Microsoft Visual Basic 2008 Professional Edition
- Microsoft Visual Basic 2008 Team System

由于后面的章节涉及一些更高级的主题,因此需要安装更多软件。另外,Visual Basic 2008 Express 版本不能用于创建 Web 应用程序和移动应用程序,也不能用于部署项目:

- 第 14 章要求安装 Microsoft Visual Basic 2008 Professional Edition 或更高版本,以便创建 Windows Forms 用户控件。
- 第 16 章要求安装 Microsoft Access 2000。
- 第 17 章要求安装 SQL Server 2005、SQL Server 2005 Express Edition 或 SQL Server 2008。

如果手头上没有这些软件,而且目前也不打算购买,那也不用担心。您仍能从本书中学到很多有用的知识。

0.4 源代码

在读者学习本书中的示例时，可以选择手动输入代码或者使用本书附带的源代码文件。本书中用到的所有源代码也可以从 www.tupwk.com.cn/downpage 和 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 框或者书名列表)，点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 978-0-470-19134-7。

下载了代码后，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(例如拼写错误或者代码错误)，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。

请给 wkservice@vip.163.com 发电子邮件，我们会检查您的反馈信息。如果是正确的，我们将在本书的后续版本中使用。

如果您在本书的勘误表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表页面上，并在本书随后的版本中更正错误。

0.6 p2p.wrox.com

如果想和作者或者其他讨论，请加入 <http://p2p.worx.com> 上的 P2P 论坛。该论坛是基于 Web 的系统，您可以发布关于 Wrox 出版的图书和相关技术的消息，与其他读者或技术人员交流。该论坛有预定功能，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>，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) 阅读使用说明，并单击 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 出版的图书相关的问题。要阅读 **FAQ**，点击任何 P2P 页面里的 **FAQ** 连接即可。

目 录

第 1 章 初识 Visual Basic 2008	1	2.5 小结	32
1.1 事件驱动编程	2	第 3 章 编写软件	33
1.2 安装 Visual Basic 2008	3	3.1 信息和数据	33
1.3 Visual Basic 2008 集成 开发环境	6	3.1.1 算法	33
1.3.1 简档文件设置页	6	3.1.2 编程语言	34
1.3.2 菜单	7	3.2 变量	35
1.3.3 工具栏	8	3.3 注释和空白	37
1.4 创建简单的应用程序	9	3.3.1 注释	37
1.4.1 工具箱	12	3.3.2 空白	39
1.4.2 改进型匈牙利表示法	15	3.4 数据类型	39
1.4.3 代码编辑器	16	3.4.1 数字类型	39
1.5 使用帮助系统	20	3.4.2 常见的整型数学运算	40
1.6 小结	21	3.4.3 整型运算的缩写方式	42
1.7 练习	22	3.4.4 字符串类型	46
第 2 章 Microsoft .NET Framework	23	3.4.5 使用日期	54
2.1 Microsoft 的信心来自于 Windows	23	3.4.6 布尔类型	60
2.1.1 MSN 1.0	24	3.5 存储变量	61
2.1.2 .NET 理念	24	3.5.1 二进制	61
2.1.3 .NET 与 Java 有些类似	25	3.5.2 位和字节	62
2.1.4 现在所处的阶段	26	3.5.3 值的表示	62
2.2 为 Windows 编写软件	26	3.5.4 值的转换	63
2.2.1 .NET Framework 类	27	3.6 方法	65
2.2.2 执行代码	28	3.6.1 使用方法的原因	65
2.3 公共语言运行库	29	3.6.2 前面曾介绍过的方法	66
2.3.1 载入并执行代码	30	3.6.3 创建方法	69
2.3.2 应用程序隔离	30	3.6.4 选择方法名称	72
2.3.3 安全性	30	3.6.5 作用域	72
2.3.4 互操作	31	3.7 小结	74
2.3.5 异常处理	31	3.8 练习	75
2.4 通用类型系统和公共 语言规范	31	第 4 章 流程控制	77
		4.1 作出决策	77
		4.2 If 语句	77
		4.2.1 Else 语句	79

4.2.2	可进行多重选择的 ElseIf 语句	80	5.4.1	创建结构	139
4.2.3	嵌套的 If 语句	81	5.4.2	为结构添加属性	142
4.2.4	单行 If 语句	82	5.5	数组列表	142
4.2.5	比较运算符	82	5.5.1	使用数组列表	143
4.2.6	字符串比较	91	5.5.2	在数组列表中进行 删除操作	146
4.3	Select Case 语句	93	5.5.3	显示数组列表中的项	149
4.3.1	不区分大小写的 Select Case 语句	96	5.6	集合	150
4.3.2	多重选择	99	5.6.1	创建 CustomerCollection 类	151
4.3.3	Case Else 语句	100	5.6.2	添加 Item 属性	152
4.3.4	Select Case 中的不同 数据类型	101	5.7	使用散列表创建查询表	154
4.4	循环	101	5.7.1	使用散列表	154
4.4.1	For...Next 循环	102	5.7.2	Remove、RemoveAt 和 Clear 方法	157
4.4.2	For Each...Next 循环	106	5.7.3	区分大小写	159
4.4.3	Do...Loop 循环	108	5.8	高级数组操作	160
4.4.4	嵌套循环	112	5.8.1	动态数组	161
4.4.5	提前退出循环	114	5.8.2	使用 Preserve 关键字	162
4.4.6	死循环	116	5.9	小结	163
4.5	小结	117	5.10	练习	164
4.6	练习	117			
第 5 章	处理数据结构	119	第 6 章	可扩展应用程序 标记语言(XAML)	165
5.1	数组	119	6.1	什么是 XAML	165
5.1.1	定义和使用数组	119	6.2	XAML 语法	167
5.1.2	使用 For Each...Next 循环	122	6.3	Windows Presentation Foundation	169
5.1.3	将数组作为参数传递	124	6.3.1	创建一个丰富的 WPF 用户界面	170
5.1.4	数组的排序	126	6.3.2	使用 WPF 通用控件	175
5.1.5	反向检索	127	6.3.3	连接事件	181
5.1.6	用值初始化数组	129	6.4	小结	185
5.2	枚举	130	6.5	练习	185
5.2.1	使用枚举	130			
5.2.2	确定状态	133	第 7 章	构建 Windows 应用程序	187
5.2.3	设置无效值	135	7.1	响应事件	187
5.3	常量	136	7.2	创建简单的应用程序	195
5.3.1	使用常量	136	7.2.1	构建窗体	195
5.3.2	各种常量类型	138	7.2.2	统计字符	198
5.4	结构	139	7.2.3	统计单词	201

7.3 创建更复杂的应用程序.....	206	8.7 FolderBrowserDialog 控件.....	261
7.3.1 Text Editor 项目.....	206	8.7.1 FolderBrowserDialog 控件的属性.....	262
7.3.2 创建工具栏.....	207	8.7.2 使用 FolderBrowserDialog 控件.....	262
7.3.3 创建状态栏.....	211	8.8 小结.....	264
7.3.4 创建编辑框.....	213	8.9 练习.....	265
7.3.5 清除编辑框.....	214		
7.3.6 响应工具栏按钮.....	217		
7.4 使用多个窗体.....	223	第 9 章 创建菜单.....	267
7.5 小结.....	227	9.1 菜单功能概述.....	267
7.6 练习.....	227	9.1.1 图形.....	267
第 8 章 显示对话框.....	229	9.1.2 访问键.....	267
8.1 MessageBox 对话框.....	229	9.1.3 快捷键.....	268
8.1.1 MessageBox 中的可用图标.....	230	9.1.4 复选标记.....	268
8.1.2 MessageBox 中的可用按钮.....	230	9.1.5 Properties 窗口.....	269
8.1.3 设置默认按钮.....	231	9.2 创建菜单.....	270
8.1.4 其他选项.....	231	9.2.1 设计菜单.....	270
8.1.5 Show 方法的语法.....	231	9.2.2 添加工具栏和控件.....	272
8.1.6 MessageBox 的示例.....	233	9.2.3 为菜单编写代码.....	273
8.2 Open 对话框.....	236	9.2.4 为 View 菜单和 工具栏编码.....	277
8.2.1 OpenFileDialog 控件.....	236	9.2.5 测试代码.....	278
8.2.2 OpenFileDialog 的属性.....	237	9.3 上下文菜单.....	281
8.2.3 OpenFileDialog 的方法.....	238	9.3.1 创建上下文菜单.....	281
8.2.4 使用 OpenFileDialog 控件.....	239	9.3.2 激活和禁用菜单项和 工具栏按钮.....	284
8.3 SaveFileDialog 控件.....	243	9.4 小结.....	287
8.3.1 SaveFileDialog 的属性.....	243	9.5 练习.....	288
8.3.2 SaveFileDialog 的方法.....	244		
8.3.3 使用 SaveFileDialog 控件.....	244	第 10 章 调试和错误处理.....	289
8.4 FontDialog 控件.....	247	10.1 主要错误类型.....	289
8.4.1 FontDialog 的属性.....	248	10.1.1 语法错误.....	290
8.4.2 FontDialog 的方法.....	248	10.1.2 执行错误.....	293
8.4.3 使用 FontDialog 控件.....	248	10.1.3 逻辑错误.....	293
8.5 ColorDialog 控件.....	251	10.2 调试.....	294
8.5.1 ColorDialog 的属性.....	251	10.2.1 创建示例项目.....	294
8.5.2 使用 ColorDialog 控件.....	252	10.2.2 设置断点.....	310
8.6 PrintDialog 控件.....	253	10.2.3 使用 Watch 窗口 进行调试.....	316
8.6.1 PrintDialog 的属性.....	254		
8.6.2 使用 PrintDialog 控件.....	254		
8.6.3 PrintDocument 类.....	254		
8.6.4 打印文档.....	255		

10.2.4	使用 Locals 窗口 进行调试	318
10.3	错误处理	319
10.4	小结	322
10.5	练习	323
第 11 章	构建对象	325
11.1	对象的概念	325
11.1.1	封装	326
11.1.2	方法和属性	326
11.1.3	事件	327
11.1.4	可见性	327
11.1.5	类的概念	328
11.2	构建类	328
11.3	复用性	329
11.4	设计对象	330
11.4.1	状态	331
11.4.2	行为	331
11.4.3	存储状态	332
11.4.4	属性	334
11.4.5	读/写属性	337
11.4.6	IsMoving 方法	340
11.5	构造函数	341
11.6	继承	343
11.6.1	添加新方法和属性	344
11.6.2	添加 GetPowerTo WeightRatio 方法	347
11.6.3	改变默认值	348
11.6.4	多态性	350
11.6.5	重写更多的方法	350
11.6.6	从对象类中继承	352
11.7	对象和结构	353
11.8	Framework 类	353
11.8.1	命名空间	353
11.8.2	Imports 语句	355
11.8.3	创建自己的命名空间	356
11.8.4	.NET Framework 中的继承	358
11.9	小结	359
11.10	练习	359

第 12 章	高级面向对象技术	361
12.1	构建收藏夹浏览程序	361
12.1.1	Internet 快捷方式和 收藏夹	361
12.1.2	使用类	364
12.1.3	搜索收藏夹	370
12.1.4	查看收藏夹	377
12.2	另一个收藏夹浏览程序	379
12.2.1	构建收藏夹任务栏图标	379
12.2.2	显示收藏夹	381
12.3	共享属性和方法	384
12.3.1	使用共享过程	385
12.3.2	使用共享方法	389
12.4	面向对象的编程和 内存管理	390
12.4.1	垃圾回收	391
12.4.2	释放资源	392
12.4.3	整理碎片和压缩	392
12.5	小结	393
12.6	练习	394
第 13 章	建立类库	395
13.1	理解类库	395
13.1.1	创建类库	396
13.1.2	为 Favorites Viewer 创建类库	397
13.1.3	多层应用程序	400
13.2	使用强名	401
13.2.1	为程序集签名	401
13.2.2	程序集的版本	403
13.3	注册程序集	403
13.3.1	Gacutil 实用程序	404
13.3.2	程序集在 References 对话框中不可见的原因	404
13.4	设计类库	405
13.5	使用第三方的类库	406
13.6	用对象浏览器查看类	407
13.7	小结	408
13.8	练习	408

第 14 章	创建 Windows 窗体	
	用户控件	409
14.1	Windows 窗体控件	409
14.2	创建和测试用户控件	410
14.3	为用户控件添加属性	413
14.3.1	添加属性	413
14.3.2	给用户控件添加方法	414
14.3.3	给用户控件添加事件	415
14.4	设计期间或运行期间	419
14.5	创建 Command Link 控件	422
14.5.1	创建 Command Link 控件	422
14.5.2	使用 Command Link 控件	430
14.6	小结	433
14.7	练习	433
第 15 章	编写自定义图形	435
15.1	创建一个简单的绘图程序	435
15.1.1	创建带有用户控件的项目	435
15.1.2	绘图程序的工作原理	436
15.1.3	GraphicsItem 类	437
15.1.4	屏幕和客户坐标	440
15.1.5	监听鼠标和绘制 GraphicsCircle 对象	441
15.1.6	使控件无效	445
15.1.7	优化绘制	446
15.1.8	选择颜色	446
15.1.9	响应单击事件	452
15.2	处理两种颜色	455
15.2.1	表明指派的按钮	457
15.2.2	使用高级颜色	463
15.2.3	使用不同的工具	466
15.2.4	绘制空心圆	467
15.3	使用图像	471
15.3.1	绘制图像	472
15.3.2	缩放图像	473
15.3.3	锁定纵横比	475
15.4	更多的图形方法	478
15.5	小结	478

第 16 章	访问数据库	479
16.1	数据库的概念	479
16.1.1	Microsoft Access 对象	479
16.1.2	表	480
16.1.3	查询	480
16.2	SQL 中的 SELECT 语句	480
16.3	Access 中的查询	482
16.4	创建一个客户查询	482
16.5	数据访问组件	486
16.5.1	DataSet 组件	487
16.5.2	DataGridView 组件	487
16.5.3	BindingSource 组件	487
16.5.4	BindingNavigator 组件	487
16.5.5	TableAdapter 组件	488
16.6	数据绑定	488
16.7	小结	494
16.8	练习	494
第 17 章	使用 ADO.NET 和 SQL Server 进行数据库编程	495
17.1	ADO.NET	496
17.1.1	ADO.NET 数据命名空间	497
17.1.2	SqlConnection 类	498
17.1.3	SqlCommand 类	499
17.1.4	SqlDataAdapter 类	502
17.1.5	DataSet 类	505
17.1.6	DataView 类	506
17.2	使用 ADO.NET 类	508
17.3	数据绑定	516
17.3.1	BindingContext 和 CurrencyManager 对象	517
17.3.2	绑定控件	518
17.4	LINQ to SQL	545
17.5	小结	549
17.6	练习	549
第 18 章	ASP.NET	551
18.1	瘦客户端体系结构	551
18.2	Web 窗体和 Windows 窗体的比较	552

18.2.1	Windows 窗体的优点	552
18.2.2	Web 窗体的优点	553
18.3	Web 应用程序基础	553
18.3.1	Web 服务器	554
18.3.2	浏览器	554
18.3.3	HTML	554
18.3.4	VBScript 与 JavaScript	554
18.3.5	CSS	555
18.4	Active Server Pages	555
18.4.1	ASP.NET Web 页面的优点	555
18.4.2	特殊的 Web 站点文件	555
18.4.3	开发	556
18.4.4	Toolbox 控件	556
18.5	构建 Web 应用程序	557
18.5.1	为客户端和服务端 处理创建 Web 窗体	557
18.5.2	VS 2008 中 Web 站点 的位置	561
18.5.3	执行数据的输入和验证	564
18.5.4	设计站点的外观	568
18.5.5	使用 GridView 构建 数据驱动的 Web 窗体	578
18.6	小结	583
18.7	练习	584
第 19 章	Web 项目	585
19.1	Web 站点验证	585
19.1.1	窗口验证	585
19.1.2	窗体验证	586
19.1.3	Web 站点管理工具	586
19.1.4	登录控件	594
19.2	小结	606
19.3	练习	607
第 20 章	Visual Basic 2008 和 XML	609
20.1	了解 XML	609
20.1.1	XML 简介	610
20.1.2	向 Visual Basic 编程 新手介绍 XML	612

20.2	地址簿项目	612
20.2.1	创建项目	612
20.2.2	SerializableData 类	613
20.2.3	载入 XML 文件	619
20.2.4	改变数据	622
20.2.5	发送 Email	623
20.2.6	创建地址列表	624
20.2.7	忽略成员	628
20.2.8	载入地址	631
20.2.9	添加新地址	632
20.2.10	地址导航	634
20.2.11	删除地址	635
20.3	集成地址簿应用程序	637
20.3.1	集成规则	638
20.3.2	从其他应用程序中 读取 Address Book	639
20.4	小结	644
20.5	练习	644
第 21 章	分布式计算与 WCF	647
21.1	Web 服务概述	647
21.1.1	Web 服务的工作方式	648
21.1.2	简单对象访问协议	649
21.2	构建 Web 服务	650
21.2.1	Web 服务示例	650
21.2.2	添加更多方法	653
21.3	了解 WCF 服务	654
21.4	小结	662
21.5	练习	663
第 22 章	使用 Windows Workflow Foundation 构建 顺序工作流	665
22.1	Visual Studio 工作流模板	667
22.2	Workflow Foundation 组件	668
22.3	顺序工作流活动	668
22.4	创建顺序工作流	668
22.5	财产税清单表的工作流	671
22.6	小结	679
22.7	练习	680

第 23 章 构建移动应用程序	681	附录 A 练习题答案	719
23.1 理解环境	681	附录 B 使用 MSF	745
23.1.1 公共语言运行库	682	B.1 软件开发生命周期	745
23.1.2 ActiveSync 和 Windows Mobile Device Center	682	B.2 MSF	746
23.1.3 Compact Framework 中 的通用类型	683	B.2.1 构想阶段	747
23.1.4 Compact Framework 类 ..	684	B.2.2 计划阶段	748
23.2 构建一个 Pocket PC 游戏	686	B.2.3 开发阶段	748
23.3 小结	697	B.2.4 测试阶段	749
23.4 练习	698	B.2.5 部署阶段	749
第 24 章 部署应用程序	699	B.3 处理平衡	749
24.1 部署	699	B.4 利用 MSF 确保成功	750
24.1.1 ClickOnce 部署	700	B.5 小结	751
24.1.2 XCOPY 部署	705	附录 C 代码安全与 SSL	753
24.2 创建 Visual Studio 2008 安装程序	705	C.1 代码访问安全性	754
24.3 用户界面编辑器	708	C.1.1 权限	754
24.4 部署不同的解决方案	711	C.1.2 安全策略	755
24.4.1 私有程序集	711	C.1.3 凭证	755
24.4.2 共享程序集	712	C.2 安全套接字层	755
24.4.3 部署桌面应用程序	712	C.3 寻找答案	756
24.4.4 部署 Web 应用程序	713	C.4 小结	757
24.4.5 部署 XML Web 服务	713	附录 D Windows CardSpace 简介 ..	759
24.4.6 有效的工具	713	D.1 使用 CardSpace	759
24.5 小结	714	D.2 向站点添加 Cardspace	763
24.6 练习	714	D.3 Internet 上的信息	764
第 25 章 如何深入学习	715	D.4 小结	764
25.1 在线资源	715	附录 E .NET Framework 版本的不同点	765
25.1.1 P2P.Wrox.com	716	E.1 .NET Framework 中的新功 能: 自 2.0 版本后添加的 主要功能	765
25.1.2 Microsoft 资源	716	E.2 .NET Framework 中的新功 能: 自 2.0 版本后添加的 小型功能	766
25.1.3 其他资源	716	E.3 Visual Studio 2008 的新功能 ..	766
25.2 脱机资源(书籍)	717	E.4 Internet 上的信息	766
25.2.1 Professional Visual Basic 2008	717		
25.2.2 Visual Basic 2008 Programmer's Reference ..	717		

第 1 章

初识 Visual Basic 2008

现在,我们已经来到了用 Visual Basic 2008 和 Windows Vista 进行编程的令人兴奋的时刻。Windows Vista 是自 Windows XP 在 2002 年第一次发布以来的首次 Windows 操作系统升级。Windows 用户界面和 Visual Basic 2008 中的重大改变使得我们很容易编写专业的 Windows 应用程序以及 Web 应用程序和 Web 服务。不要犹豫,请赶紧升级为 Windows Vista 吧, Visual Basic 2008 也可用于为 Windows XP 编写专业的应用程序。

即使您以前从未编写过任何程序,通过本书的学习仍然可以快速掌握 Visual Basic 2008 编程语言。我们将循序渐进地介绍 Visual Basic 2008 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 2008 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 2008 是一种简单的、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 2008 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各种问题。

Visual Basic 2008 可以帮助您创建运行于 Microsoft Windows 操作系统(如 Windows Vista、Windows Server 2008 和 Windows Mobile 6)下的应用程序。如果您正在看这本书,就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 2008 语言的方方面面,以及 Microsoft 的 .NET Framework 的基础知识。您会发现,其实它比我们想像的要容易。在了解这一点之前,您将感受到可以轻松用 Visual Basic 2008 创建各种不同的程序。

Visual Basic 2008 可用于创建 Web 应用程序和 Web 服务,以及可在 Pocket PC 和 SmartPhone 上运行的移动应用程序。然而,在扩展到其他平台之前,您应从学习 Windows 应用程序开始。

本章的主要内容如下:

- 事件驱动编程
- Visual Basic 2008 的安装
- Visual Basic 2008 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

1.1 事件驱动编程

Windows 程序不同于过去的 MS-DOS 程序。DOS 程序自始至终遵循一种相对严格的路径。虽然这不会限制程序的功能,但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,必须穿过这条走廊,且必须清除途中可能会遇到的所有障碍,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:单击一个按钮、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,就可以直接向新的目的地出发,而不用再返回到起点。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。

在 Windows 下编程的另一大优势是可以把硬件抽象化。这就是说,Windows 很注意与硬件的通信。我们不需要知道市场上每种激光打印机的内部工作原理,只管输出结果即可;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,只有经过严格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 Visual Basic 语言的使用率远高于其他任何一种语言。

Visual Basic 取消了用户界面(UI)繁琐的代码编写工作,从而使 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 已在原来的基础上得到了扩展。第三方供应商也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,就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 3.0 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。Microsoft 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 又被升级为 Visual Basic 4.0 和 Visual Basic 5.0。最重要的是,它们使开发人员更容易编写代码,而这些代码又能被其他语言的开发人员所使用。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。Microsoft 公司开发 ADO,是为了帮助 Web 开发人员使用 Active