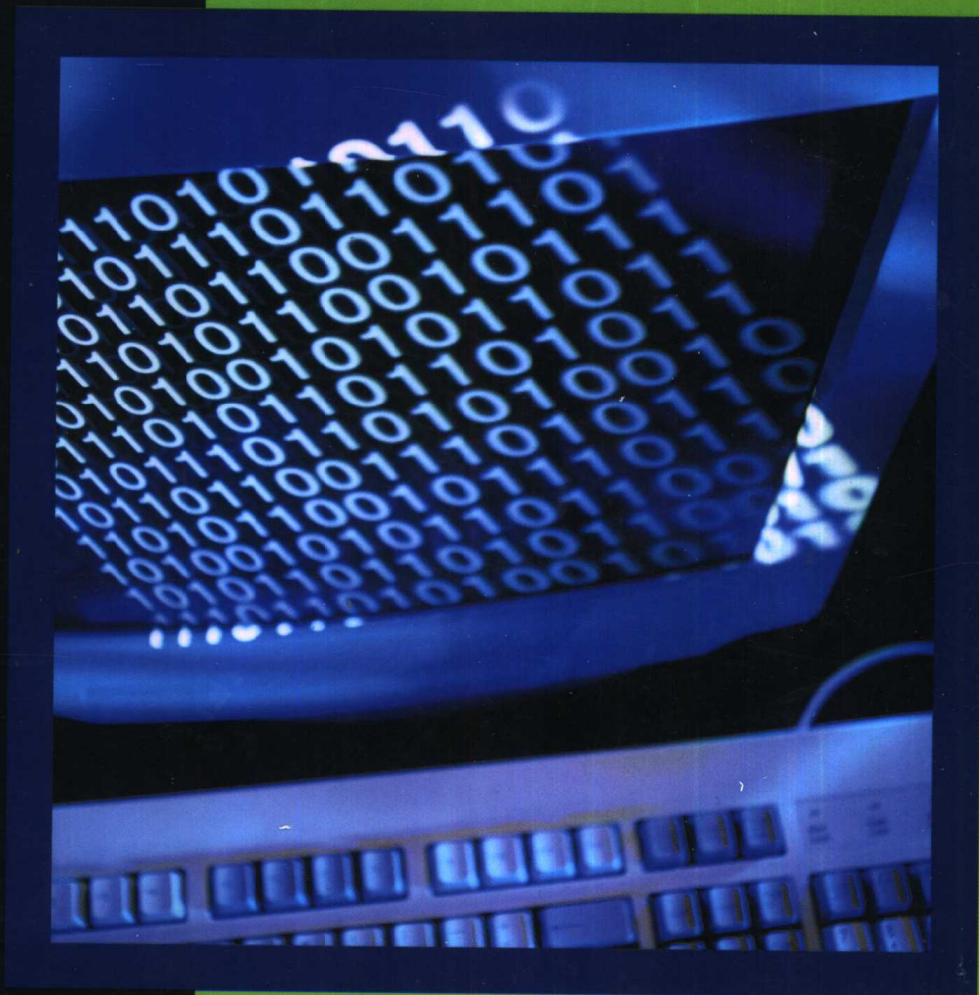


Visual Basic.NET Programming



Visual Basic.NET

编程指南

使用Visual Basic.NET漂亮而强大的编程环境，
编写极具可读性的面向对象的应用程序

〔美〕 Harold Davis 著

战晓苏 等译



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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



Visual Basic.NET

编程指南

Microsoft Visual Basic .NET 2002 中文版

Microsoft 中国有限公司 出版

ISBN 7-302-06100-3

定价：39.00元

06100-3

清华大学出版社

北京 2002 年 10 月第 1 版第 1 次印刷

06100-3



Visual Basic.NET Programming

Visual Basic.NET

编程指南

〔美〕 Harold Davis 著

战晓苏 等译

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 提 要

本书以Visual Basic.NET为主线,对Visual Basic语言的应用开发做了详尽细致的介绍。本书共20章,主要介绍.NET Framework、Visual Studio集成开发环境、Windows窗体应用程序的开发及其控件和组件、Web应用程序的开发及其控件和组件、面向对象的程序设计(OOP)、程序调试、XML和安装部署各种应用程序。本书附录还介绍了VB.NET的新型帮助系统,以及VB6到VB.NET应用程序的升级。本书的特点是内容新颖、深入浅出,并且提供了大量的示例项目和对象源代码,便于读者理解掌握Visual Basic.NET的应用技巧。

本书既可以作为Visual Basic.NET的入门教材,也可以作为程序员的培训教材,是广大计算机爱好者应具备的实用工具。



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: 01-2002-0903

图书在版编目(CIP)数据

Visual Basic.NET编程指南/(美)大卫(Davis, H.)著;战晓苏等译.一北京:电子工业出版社,2003.1

书名原文: Visual Basic.NET Programming

ISBN 7-5053-8030-3

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

中国版本图书馆CIP数据核字(2002)第074640号

责任编辑:李莹

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:25.125 字数:640千字

版次:2003年1月第1版 2003年1月第1次印刷

定价:42.00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

献给我的父母——马丁和维吉尼亚

致 谢

他们说养大一个小孩需要很多人付出辛劳。无疑，创作这样一本书同样需要一个团队。如果没有**Sybex**公司的出色员工，就不会有这本书的出版。为此，我要向他们一一表达我的谢意：**Richard Mills**和**Denise Santoro Lincoln**，创办了这个项目并邀请我加盟；**Denise**，启动了这本书的策划并推动它从构想直到初具雏形；**Mae Lum**，担负起了出色的逻辑组织工作，使本书渐渐成熟完备起来；**Scott Swigart**，非常认真地从技术角度校阅了我的原稿，他的**.NET**技术知识极大地丰富了这本书；**Marilyn Smith**，负责编辑加工稿件，他切实的工作使本书成为现在这个令人骄傲的样子。

我还要感谢**Sybex**出版小组，包括插图制作**Tony Jonick**；电子出版专家**Judy Fung**；校对员**Nancy Riddiough**、**Nelson Kim**、**David Nash**和**Nanette Duffy**；以及索引员**Jerilyn Sproston**。

除了**Sybex**的员工以外，我还要感谢我的朋友和代理人，**Matt Wagner**和**Bill Gladstone**，他们俩都是**Waterside Productions**公司的，没有他们这本书也不会存在。同时，**John Kilcullen**和**Stephen Compagni Portis**也间接地对本书的创意做出了贡献。

最后，我要感谢我的妻子**Phyllis Davis**，她又一次为我的书做了主要贡献。是**Phyllis**和我的孩子，**Julian**和**Nicholas**，给了我创作这本书的灵感、空间和动力。

译者序

Visual Studio.NET是一套完整的开发工具，用于生成ASP Web应用程序、XML Web服务、桌面应用程序和移动应用程序。Visual Basic.NET、Visual C++.NET和Visual C# .NET全都使用相同的集成开发环境（IDE），该环境允许它们共享工具，并有助于创建混合语言解决方案。另外，这些语言利用了.NET Framework的功能，此框架提供对简化ASP Web应用程序和XML Web服务开发的关键技术的访问。

.NET框架是一种新的计算平台，它简化了在高度分布式Internet环境中的应用程序开发。
.NET框架旨在实现下列目标：

- 提供一个一致的面向对象的编程环境，而无论对象代码是在本地存储和执行，还是在本地执行但在Internet上分布，或者是在远程执行。
- 提供一个将软件部署和版本控制冲突最小化的代码执行环境。
- 提供一个保证代码（包括由未知的或不完全受信任的第三方创建的代码）安全执行的代码执行环境。
- 提供一个可消除脚本环境或解释环境的性能问题的代码执行环境。
- 使开发人员的经验在面对类型大不相同的应用程序（如基于Windows的应用程序和基于Web的应用程序）时保持一致。
- 按照工业标准生成所有通信，以确保基于.NET框架的代码可与任何其他代码集成。

Visual Basic.NET就是在以上环境和平台支持下的程序设计语言。本书从最基本的知识开始，介绍了.NET框架以及Visual Studio开发环境的使用方法和技巧，进而详细介绍了VB.NET里的各种常用的编程元素及其属性、方法和事件，包括窗体、控件和组件。同时还介绍了一些编程基本技术，包括事件驱动模型编程、OOP（面向对象的程序设计）、ADO.NET、ASP.NET、XML等等。最后介绍了应用程序的安装和部署。在附录中还介绍了旧版本的VB与VB.NET的不同点以及应用程序的升级。

另外，本书是一本从基础开始学习VB.NET的优秀参考书，学习完本书后，你将成为一名合格的VB.NET程序员。

本书由战晓苏、任鸿翻译并统稿。参加翻译和审校工作的还有杨涛、潘征、余平、石雄、李德勇、杨秀合、张敬松、杨天梁、张志刚、刘世华、邹德新、薛林光、王友军、华书重、刘庶忠、张华勋。

在本书的翻译过程中难免有所疏漏，请读者批评指正。

引言

最初，.NET只是我意识里的遥远回响。在很长一段时间里——似乎有好几年，实际上确实是好几年——我一直听到有关.NET的十分喧嚣的宣传，但却很少关心这个东西。因为我主要关心的是可靠且高质的编程工具，而不是使用新奇的玩意，并且我对这种推广活动持怀疑态度；同时相信，VB6对我以及我的咨询客户和我的读者来说已足够用了。直到我尝试着使用了.NET并证明了它的可靠性和实用性之后，我才开始花时间来了解.NET。

什么是.NET策略

在我们说出“啊哈”之前，尽管我已经对.NET和Visual Basic.NET有了一些领悟，但还是愿意停下一会儿，考虑.NET到底是什么又不是什么。显然它有点让人感到迷惑，因为微软所说的.NET有些只能称之为“幻想”，有些是市场术语，而另外一些东西——比如Visual Basic .NET——现在确实已经成为很扎实的技术（意思是说，还有什么比编程语言更扎实的技术呢？）。

抛开这些让人头晕的术语，微软的.NET策略包含以下三方面：

- .NET Framework和Visual Studio.NET，一种运行时平台和包括Visual Basic.NET语言在内的开发环境。
- .NET企业服务器，一组根据市场目标而加上.NET名字的企业服务器产品，比如，Biztalk Server、Exchange Server、Mobile Information Server和SQL Server。
- .NET我的服务（MyServices），有时也叫Hailstorm，用来创建服务，比如为许多分布式应用程序所用的护照（Passport）用户验证服务。

本书首先介绍依靠.NET Framework的运行时平台，在Visual Studio开发环境下，使用Visual Basic.NET语言编写应用程序。可以认为.NET应用程序的运行时平台和Java应用程序运行的“沙盒”（sandbox）类似，除此之外，.NET Framework主要部署在微软的Windows平台上。

Visual Basic.NET的特点

那么，这样一种不同的编程语言和开发环境有什么激动人心的地方呢？从某些方面来看，这跟谚语所说的“布丁好坏，一尝便知”是一个道理，你需要真正拿它来做个测试并且自己感觉一下。同时，我愿意与你分享一些我看重的和注意到的功能。

我发现的第一个非常特别的特点就是，Windows应用程序可以在Internet Explorer浏览器里运行（当然，也有一些要求：目标系统需要安装.NET Framework和最新的Internet Explorer浏览器）。每一次在Visual Studio开发环境下成功运行一个程序，比如Windows应用

程序，就要创建一个编译过的可执行文件。如果在Internet Explorer中打开这个文件，它可以像普通Windows应用程序一样，在全世界都可以运行、查看和工作。这为人们利用非常丰富的用户界面轻松地创建分布式应用程序创造了可能，而且用户除了点击超级链接之外，无需再做任何事（微软称之为无影响（no-touch）部署）。

下一个特点涉及到有点傻的编程想法。应用程序窗口只能是方型的问题经常困扰我。现在不用了！在第6章“使用窗体方法和模块”中，将演示怎样创建椭圆形的窗口。或许我们不需要创建使用曲线界面的应用程序，但是起码应知道窗口不一定是方的。

Visual Basic.NET另外一个让人激动的地方是，它允许使用事件驱动编程模式来创建Web应用程序（而不是使用从上到下的线性HTML模式）。通过滚动自己的Web服务，只需单击一些按钮就可以完成XML定义和查找文件。第19章“构建ASP.NET Web应用程序”将解释创建Web应用程序和Web服务的技巧。

本书的类型

在这里和本书各章中提到的许多功能听起来有些悬，但是，本书会像农夫种地一样，脚踏实地地进行介绍，因此读者看不到那些“哎呀，难道这个技术不是很神奇吗？”之类的惊呼。本书所述内容都是有事实根据的，它只是要告诉大家应该怎么做。

作为本书的作者，我向读者保证的是我不会告诉大家任何我没有试过（并且着手做）的事。书中的每一个例子、每一行代码，都是我自己的成果（有时简单明了，有时又令人心烦）。我相信，如果我是一个编程的超级英雄，会给大家更好的指导。显示我知道的比别人多并不是我的工作，我的工作只是帮助各位找到自己的方法。有我和你在一起，会让你避免进入陷阱，并一步一步地向前走，从黑暗中找到光明。我知道本书中的每一个例子都能工作，因为我已经亲自试过了。

本书并未讲述怎样从Visual Basic 6变迁到Visual Basic.NET的历史，也没有探讨它的未来，而只是介绍了现在的Visual Basic.NET。可以说这是一本实用的指导书。

本书的阅读对象

本书不要求使用这本书的读者必须了解任何形式的编程语言，但我强烈建议，各位最低限度应当对微软的Windows操作系统很熟悉，有一定的HTML基础，可以写一两个脚本。如果你是专业或高级程序员，可以把本书作为学习.NET的基础。如果你已经对.NET很有经验，则本书可能不太适合你。

本书面向的是对下列这些问题做肯定回答的读者：

你是否曾经想学编程？如果想学编程，我相信Visual Basic.NET是最好用的、最简单的语言。本书可以帮助你一步一步，轻松地学会它。

你是否对.NET编程中近期出现的“新的、新事物”等说法感到好奇？如果想理解使用.NET Framework编程的具体细节及相关知识，可以从这本书中找到这样的信息。

你是否对以前版本的Visual Basic有一定的经验，比如VB6.0？如果确实的话，本书将帮助你转变观念，从而快速创建强大的Visual Basic.NET应用程序。

你是否是一个初学者或中级开发者，主要关心Visual Basic.NET基于任务的使用指导？如果是这样的话，本书也适合你，可以根据本书讲述的步骤快速地完成任

务。我为Visual Basic.NET那漂亮而强大的编程环境感到振奋。我希望不管你的编程水平有多高，都能从这本书中找到感兴趣的有用信息。

怎样下载代码

本书的大部分例子都不是很长，它们只用于强调应该怎样做，而不是完整的运行资料或软件作品。我鼓励大家根据这些例子重新创建项目里的对象，并自己用键盘输入源代码，这样各位会学到更多的东西。

当然，也可以下载包含本书使用项目的源代码。如果手工输入的代码不工作，可以用它来做比较。Sybex在它的网站（<http://www.sybex.com/>）上发布了本书的所有代码。可以使用标题、作者或ISBN号4038来搜索本书，并点击Download按钮进行下载。但只有读者接受了许可协议，才能下载列出的本书代码，且这些代码是以章为单位按压缩文件格式组织的。

怎样和作者联系

我尽量使本书有用和正确。请让我知道各位的想法。我为本书特别设置了一个电子邮件地址：vbnet@bearhome.com。我很感谢大家告诉我对本书的任何建议和使用本书过程中遇到的错误。

希望大家喜欢这本书。Visual Basic.NET是令人激动的、极好的、更容易使用的编程语言。本书的目标就是帮助读者快速地学会它。

目 录

第1章 理解Visual Basic.NET	1
什么是.NET	1
.NET Framework中的程序设计	5
.NET语言	9
使用文本编辑器创建VB.NET程序	12
Visual Studio.NET的要求	14
小结	15
第2章 项目、窗体和按钮介绍	16
理解项目和解决方案	16
第一次使用	18
配置项目和环境	21
添加Windows窗体	25
运行项目	32
添加控件	34
介绍事件处理程序	35
小结	39
第3章 构建用户界面	40
显示消息框	40
添加窗体控件	50
使用If语句	60
铺设窗体	61
小结	67
第4章 使用Windows窗体属性	68
使用窗体属性	68
设置窗体的样式	70
设置默认按钮	72
设置背景色和前景色	74
设置字体	80
分配图标	82
设置窗体的大小和位置	83
指定Text属性	85
控制窗体的外观和行为	86

设置动态属性	86
小结	88
第5章 窗体的事件和生存周期	89
理解事件驱动的程序设计	89
处理事件	90
监视事件	92
在Load事件中初始化窗体	96
理解Event语法	99
避免事件无限循环	100
为多个事件指定一个事件处理程序	100
小结	101
第6章 使用窗体方法和模块	102
使用窗体方法	102
使用代码模块	104
使用窗体	108
使用图形对象	117
创建MDI应用程序	123
小结	126
第7章 使用Windows窗体控件	127
使用工具箱	127
使用PictureBox控件	131
添加CheckBox控件	136
使用GroupBox控件	139
使用ComboBox、ListBox和CheckedListBox控件	143
使用ErrorProvider组件	151
小结	153
第8章 添加StatusBars、ToolBar、ToolTip和选项卡	154
使用StatusBar控件	154
使用ToolBar控件	157
添加ToolTip组件	162
添加选项卡	163
小结	165
第9章 使用公用对话框	166
使用公用对话框控件	166
使用ColorDialog控件	167
使用FontDialog控件工作	169

使用SaveFileDialog和OpenFileDialog控件	171
使用文件	175
小结	180
第10章 使用Timer组件、EventLog和服务Controller类	181
使用Timer组件	181
使用服务器资源管理器	188
编写事件日志	188
使用ServiceController类	193
小结	197
第11章 创建菜单	198
使用MainMenu控件	198
添加上下文菜单	205
使用复选菜单项	206
在代码中操作菜单	207
查看菜单后面的代码	209
以交互方式创建菜单	212
小结	216
第12章 应用程序中的打印	217
使用打印组件和控件	217
打印一页	218
生成简单的打印演示	219
小结	224
第13章 错误、异常和调试	225
理解错误和异常	225
过时的语句: On Error GoTo	228
理解结构化异常处理	230
测试与调试	239
小结	246
第14章 使用对象浏览器	247
打开对象浏览器	247
使用对象浏览器界面	250
浏览命名空间	252
浏览类	253
小结	254
第15章 VB.NET中的面向对象编程	255
理解面向对象编程	255

创建类	258
生成类库	261
使用类的命名空间调用类	262
使用类模块	266
使用类的属性、方法和事件	267
创建集合类	275
小结	279
第16章 创建Windows控件	280
创建继承的控件	280
创建复合控件	286
小结	291
第17章 使用数据和ADO.NET	292
数据库基础回顾	292
使用数据组件	293
使用数据组件创建数据集	297
绑定数据到控件	306
小结	312
第18章 使用XML	313
理解XML	313
查看XML模式	314
读取XML数据到数据集	316
读写XML内容	321
小结	330
第19章 构建ASP.NET Web应用程序	331
创建Web窗体应用程序	331
添加控件到Web窗体	343
使用Web服务	350
小结	355
第20章 部署应用程序	356
介绍无影响部署	356
选择安装和部署项目的类型	358
添加安装项目到Windows应用程序	360
使用安装向导创建Web安装项目	364
小结	369

附录A 使用VB.NET的帮助系统	370
附录B 从VB6到VB.NET迁移应用程序	376
附录C VB6和VB.NET之间语法关键字的不同	380
附录D VB.NET关键字	384
附录E 可选的VB.NET的命名空间和类	386

第1章 理解Visual Basic.NET

- .NET Framework
- .NET语言
- 命令行编译
- Visual Studio.NET的硬件要求

本章将介绍与Visual Basic.NET (VB.NET) 相关的重要背景资料。如果读者急于直接进行编程的话, 则不需要从本章开始阅读(但是你可能需要对照本章的“Visual Studio.NET的要求”一节, 确认已经具备了所需的全部工具)。可以跳过本章直接阅读第2章“项目、窗体和按钮介绍”(第2章中的例子程序将带领读者建立并运行一个一闪而过的Windows程序)。当继续学习VB.NET时, 深入理解VB.NET的概念是非常重要的, 因此读者可以再回到这一章重新了解它的背景资料。

另一方面, 在使用VB.NET编程以前, 如果读者愿意循序渐进地学习, 那么可以从本章开始。掌握本章介绍的概念, 更有助于你成为一名合格的VB.NET程序员。

本章包括一项重要的基本内容: 命令行编译。我们可能试图不使用Visual Studio集成开发环境(Integrated Development Environment, IDE)来创建窗口或者Web程序。这样做虽有点傻, 但是重要的是要大家知道这样做是可以的, 而且解决方法不仅仅只有Visual Studio一种(有关Visual Studio的介绍贯穿了本书其他所有章节)。大家应该清楚地意识到VB.NET语言不能等同于Visual Studio.NET的集成开发环境, 它只是其中的一个组成部分。

什么是.NET

由于经常在不同的上下文中使用, 术语.NET的含义似乎有些模糊。这些上下文大部分是市场宣传性质的语言, 而不是在理解科技概念时使用的精确语言。下面列出使用.NET的一些场合:

- 用来指代.NET Framework, 一个主要由类库组成的运行时平台和编程框架
- 包括Visual Studio.NET, 一个专业的优化集成开发环境, 以便更好地使用VB.NET和C#.NET等.NET Framework语言进行工作
- 用来指代.NET Enterprise Server, 即企业服务器产品的一个程序集, 诸如Biztalk Server、Exchange Server、Mobile Information Server以及SQL Server等等, 它们为.NET概念赋予了市场价值
- 用来描述.NET的个性化服务(有时也称为Hailstorm), 即一系列创建服务的操作, 例如清单、合同、进度信息等等, 这些服务都可以通过与平台和语言无关的方式来访问

本书主要讨论的是上述场合中的前面两个含义：.NET Framework和Visual Studio.NET。本章讨论的重点是对.NET Framework的理解。

.NET Framework

.NET Framework由两个主要部分组成：

- 通用语言运行时库（Common Language Runtime, CLR），即管理代码执行的平台（将在下一节详细讨论）
- .NET Framework类库

.NET Framework的这两个组成部分与操作平台以及Visual Studio.NET的关系如图1.1所示。

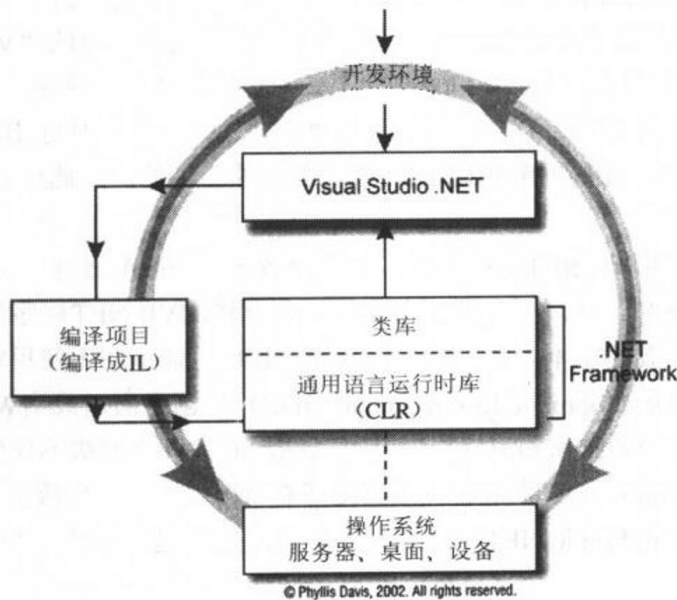


图1.1 .NET Framework的这两部分有不同的功能：Framework的类库用于构造应用程序，而处于操作系统上端的CLR层用于管理程序的执行

阅读到本章“使用文本编辑器创建VB.NET程序”时可以看到，不使用Visual Studio.NET也能够创建、编译以及运行VB.NET程序。但是随着程序复杂性的增加，问题就变得比较麻烦了。除了本章中给出的一个例子以外，本书还会阐述怎样使用Visual Studio来创建VB.NET应用程序，因为这是解决问题的最简单方法。但重要的是还必须认识到Visual Studio对于.NET Framework定义来说并不是它的全部。

.NET Framework的基类库是类型、类以及接口的庞大程序集，这些类型、类以及接口组成了应用程序的基础，构成了应用程序的蓝图，它们能用于创建一些应用程序，包括Windows窗体、Web表单以及Web服务应用程序。.NET Framework类型是创建.NET应用程序、组件和控件的基础，它能够执行多项功能，例如描述值的类型、执行输入/输出（I/O）操作以及提供数据访问功能。大多数程序员都熟悉的一种值类型是整型（Integer），在.NET Framework中整型被称为Int32，它指的是包含32位带符号整数。

说明: .NET语言可以混合使用, 即一种语言可以使用其他语言编写的类库, 例如, 一个VB.NET的程序可以使用由C#.NET语言开发的类, 反之亦然。为了确保这种互用性, .NET Framework的类型必须遵守通用语言规范(Common Language Specification, CLS)。任何遵循通用语言规范的语言都可以混合使用。

.NET Framework使用“点”操作符(.)来指明层次。相关的类型组织成一个命名空间, 这样可以更便于查找(更详细的内容请见本章的“命名空间”一节)。

从左到右阅读直到碰到第一个点分隔符时, 得到类型的第一部分, 即命名空间的名称。从右到左阅读直到碰到最后一个点分隔符时, 得到类型的最后一部分, 即类型的名称。例如System.Boolean描述了在System命名空间中的一个Boolean型的变量值。System.Windows.Forms.MessageBox描述了在Forms命名空间中的一个MessageBox型的类, Forms同时又是Windows命名空间的一部分, 而Windows则又是System命名空间的一部分。

提示: 当指代System命名空间中的一个成员时, 通常可以省略System。例如, 将变量声明为Dim IsEmployee As Boolean, 等同于将变量声明为Dim IsEmployee As System.Boolean。

由上面的例子可以看出, System命名空间是.NET Framework中所有类型的根命名空间。所有的应用程序所使用的基本数据类型都包含在System命名空间或Microsoft命名空间中。

System命名空间最重要的类型之一是System.Object类型, 也称为Object类, 它位于.NET类型层次结构的根部, 是.NET Framework中所有类最终的父类(或称为超类)。这就意味着Object类的成员, 例如GetType()和ToString()都包含在所有的.NET类中。

说明: 不要被对象、类型、类和层次等概念给弄糊涂了。开始编写程序时, 这些概念基本上都是通过相当直观的方式来使用的。但是如果哪位想直接跳到后面的章节进行阅读, 可以参考第14章“使用对象浏览器”, 这一章介绍了对象浏览器, 它是学习类层次结构的最好工具。关于面向对象的编程概念以及本章中用到的一些术语, 请参见第15章“VB.NET中的面向对象编程”中的详细介绍。

公共语言运行时库

CLR管理着编译过的.NET程序的执行情况。CLR的角色类似于Sun公司的Java虚拟机(Java Virtual Machine, JVM)以及旧版VB中封装的VB运行时库。

CLR是所有.NET语言的运行时库, 它的功能是执行和管理以任何能够用于.NET平台的语言编写的所有代码。

在.NET平台上编写应用程序时, 可以从许多语言中选择其中的一种。最常用的语言是VB和C#(读音为“see sharp”), 也可以使用COBOL、Eiffel、Pascal甚至Java语言来创建.NET应用程序。每一种语言都有自己的编译器, 它是由第三方语言编写而成的。编译器通常会将程序代码编译为机器代码, 但是, 特定于语言的实时编译器(just-in-time, JIT)会将VB、C#或者其他任何语言编写的代码编译为一种名为微软中间语言(Microsoft Intermediate Language, MSIL, 简称为IL)的语言。实际上使用和执行的就是这种IL语言。

在执行代码的过程中, CLR使用另一种编译器将IL语言编译为特定于程序运行平台的机器代码。就是说, CLR既可以将IL语言编译为在AMD处理器上运行的代码, 也可以将其编译为在Pentium处理器上运行的代码。