

微软.NET程序设计系列

Microsoft Press

- ◆ 欧美读者评价★★★★★
- ◆ 微软权威编程专家精心编著
- ◆ 入围欧美电脑图书排行榜
- ◆ 大量可重用的范例代码
- ◆ 赏心悦目的版式设计
- ◆ 附开发职业规划蓝图

MICROSOFT

VISUAL BASIC .NET

MICROSOFT

和 VISUAL C# .NET

面向对象程序设计

[美] Robin A. Reynolds-Haertle 著

孙瑛霖 丁建扬 译  
Visual Studio .NET产品组 审校

Microsoft  
.net



清华大学出版社

微软.NET 程序设计系列

# Visual Basic .NET 和 Visual C# .NET 面向对象程序设计

美] Robin A. Reynolds-Haertle

孙瑛霖 丁建扬

Visual Studio .NET 产品组

著

译

审校

清华大学出版社

(京)新登字 158 号

## 内 容 简 介

本书是《微软.NET 程序设计系列》丛书之一, 主要介绍用 Microsoft .NET 框架中功能最为强大的两种开发语言: Visual Basic .NET 和 Visual C# .NET 进行面向对象编程。本书共分 14 章, 分别介绍了面向对象中常用的基本概念、类方法的使用、继承的使用、事件和异常对更改的响应、接口的创建和实现、多态的用法等。本书语言简洁、易懂, 结构清晰, 非常便于读者理解和学习面向对象编程语言。

本书适用于 Visual Basic .NET 和 Visual C# .NET 初学者。

**OOP with Microsoft® Visual Basic® .NET and Microsoft Visual C#™ .NET**

**Microsoft Press**

**Copyright © 2002 by Microsoft Corporation**

**Original English language edition published by Microsoft Press, a Division of Microsoft Corporation**

**All rights reserved.**

**No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher. For sale in the People's Republic of China only.**

本书中文简体版由 Microsoft Press 授权清华大学出版社出版发行, 未经出版者书面许可, 不得以任何方式复制或抄袭本书的任何部分。

## 图书在版编目 (CIP) 数据

Visual Basic .NET 和 Visual C# .NET 面向对象程序设计 / (美) 哈尔特编著; 孙瑛霖, 丁建扬译. —北京: 清华大学出版社, 2002

书名原文: OOP with MS VB.NET and VC#.NET

ISBN 7-302-05640-4

I. V... II. ①哈...②孙...③丁... III. ①BASIC  
语言—程序设计②C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 045599 号

北京市版权局著作权合同登记号: 图字 01-2000-2100 号

**版权所有, 翻印必究。**

**本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。**

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 杨志娟

印 刷 者: 北京牛山世兴印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×960 1/16 印张: 22 彩插页: 4 字数: 492 千字

版 次: 2002 年 7 月第 1 版 2002 年 7 月第 1 次印刷

书 号: ISBN 7-302-05640-4/TP·3324

印 数: 0001~6000

定 价: 36.00 元

# 《微软.NET 程序设计系列》序

英文版 Visual Studio .NET 是 2002 年 2 月 13 日发布的，之所以选择这一天，原因很多，但其中一个很浪漫的因素是第二天是情人节，比尔想把 Visual Studio .NET 作为礼物送给全世界的程序员。

摩尔定律告诉我们，每过 10 年计算机系统性能将会提高 100 倍，通信带宽也会增加 100 倍，而需要的资金却不会增加。在过去的 30 年中，这种情况确实发生了。由于微软和计算机界的共同努力，PC 进入了家庭，软件也变成了一个巨大的产业。软件开发在过去的数十年里发生了巨大的变化，持续不断的技术进步和创新推动软件工业发展到了一个新的阶段。

这期间，软件开发工具所起的巨大作用是有目共睹的。早期的程序开发，所使用的编程语言是机器指令，编程人员只能使用二进制代码编写程序。由于硬件条件的限制，在编写程序时考虑最多的是如何节省资源，尽可能少的使用 CPU 时间和内存。因此，程序开发难度之大是当今年轻的程序员们难以想象的。后来，技术人员发明了程序设计语言，从汇编语言到高级程序设计语言，包括结构程序设计语言，逻辑/函数程序设计语言和面向对象的程序设计语言。尤其是软件危机所引发的软件工程的革命，软件重用、组件开发等大量新技术的涌现，软件开发的效率大大提高了，同时也大大增强了软件的功能。

大家可能还记得程序设计从字符模式转换到图形界面的情形。那时硬件的速度依然很慢，人们所习惯的是 MS-DOS 的编程规则和 MS Basic、Turbo Pascal 和 Quick Basic 等编程工具。由于没有与图形界面相适应的开发工具，软件开发变得十分困难。幸运的是，Visual Basic、PowerBuilder、Delphi 等几种突破性开发工具的诞生很快就改变了这种状态，使得图形界面下的应用程序开发和字符模式下的开发一样简单。

Internet 的诞生和流行使软件设计发生了质的变化。运行在服务器上的应用程序将结果以 HTML 格式发送给浏览器，而客户端可以随时随地连接到任意一个站点，得到想要的信息。应用程序不需要同时存在于服务器和客户机。主要的应用程序都运行在服务器上，客户机只需安装浏览器。大量的软件开发工具迅速地应运而生。随着大量 Web 应用程序的开发，软件设计人员意识到，仅在一端(服务器端)有智能是不够的，人们真正需要的关键应用程序应该是两端都具有智能，即让不同的计算机针对同一问题协同工作。应该让用户看到来自许多不同 Web 站点的信息，同时容许用户对这些信息进行编辑和加以注释。

事实上，其典型应用即是电子商务。那么，如何为一个复杂的机构建立电子商务呢？比如说，如何将 A 公司连接到 B 公司？A 公司内部可能运行着许多不同的系统，有大量的员工需要与不同的公司分别从事不同的业务。而那些公司的情况也同样复杂。如何来设计这样的软件，使得有关双方可以通过数字化的方案了解彼此的进展和状态，从而与当今基于电话和纸张的沟通相比，大大地提

高工作效率呢？

这就是分布式计算。分布式计算的基本思想是将不同的系统互联互通，这在学术界已经存在很长时间了。今天，分布式计算已成为一种实实在在的需求。因此，必须有一种有效的开发工具来帮助开发人员建立这种分布式计算系统。

1996年，微软公司和业界其他公司一起，将 SGML 结构化文档标准归纳、概括为一种新的标准——XML。这一标准使数据共享获得新的动力，但仅有 XML 还不能实现互联网上的计算机之间的“真正”相联——不能发现其他计算机所提供的服务，不能彼此发送消息，通过消息共享服务。现实的情况是，计算机之间的连接不一定可靠，双方的软件开发人员也未协调过彼此的开发，甚至另一端的计算机上可能正运行着恶意代码。即便面对这样的现实，开发人员依然必须确保他们的计算机安全可靠。这是一个巨大的挑战，一个计算机行业必须齐心协力方能面对的挑战。

迎接挑战的武器是 XML Web 服务，即建立完全基于开放标准的服务。这些标准确保在不同的系统上生成相同格式的 HTML 文件。鉴于 XML 标准已得到广泛认可，更大的竞争变成谁能以最优的方式实现这些标准。如同在图形界面世界有 Macintosh、OS2、Windows 和其他系统在竞争。众多计算机公司将竞相建立各种 Web 服务，而且这些 Web 服务是可以互操作的。为了避免 Web 服务不兼容的问题，微软公司联合 IBM 及其他在全球计算机界最有影响的 50 家公司一起组成了 WSI，即 Web 服务协调工作组织，共同制定有关 Web 服务的标准。由于这些公司在全球的代表地位和影响，从而确保了基于这些标准的 Web 服务能够很好地协调工作。

这种真正的分布式计算需要新的开发工具来支持，而它不应该只提供一种编程语言。更为重要的是，用户可以不必重新建立他们已有的软件应用系统。

.NET 不仅仅表现为 Web 服务，其根本的目的是通过 XML Web 服务使人们可以在任何时间、任何地点、通过任何设备获取所需要的信息。通过 XML Web 服务实现计算机与手持设备、车载设备、电视等家用电器等的互联互通。

让计算机编程变得更容易，为全球软件开发人员提供功能更加强大、更加好用的开发工具，是微软公司长期坚持不懈为之努力的一项使命。Visual Studio .NET 正是基于这种蓝图而设计的，它为开发下一代以 XML Web 服务为中心的应用程序而设计，是软件开发有史以来功能最强大、最受欢迎的软件开发工具。它有许多激动人心的新功能，提供统一的集成开发环境，支持在同样的开发环境里用 Visual Basic、Visual C++、Visual C#、Visual J# 和其他数十种编程语言，编写、调试和部署各种应用程序。Visual Basic 进化为完全面向对象的编程语言；Visual C# 作为一种新的编程语言集中了 C 和 C++ 的优点；Visual C++ 在保留原有十分强大的编程能力的基础上，增加了开发托管代码 (Managed Code) 功能。2002 年初，C# 和 CLI (Common Language Infrastructure, 通用语言架构) 已被批准为 ECMA (欧洲计算机制造协会) 的标准。

正是基于这些实际的需求，经微软独家授权，清华大学出版社引进并隆重推出了《微软 .NET 程

序设计系列》丛书。本丛书从最基础的知识讲起，使用浅显的语言、生动的实例演示，全面而系统地介绍了如何使用 Visual Studio .NET 和 .NET 框架包含的各种功能。旨在循序渐进地带领想了解、掌握 .NET 技术的读者进入 Visual Studio .NET 的美妙世界。一书在握，您将发现，程序开发原来是如此轻松且充满乐趣！

到目前为止，《微软.NET 程序设计系列》共包括 6 本书：

- 《Visual Basic .NET 程序设计》

介绍了 Visual Basic .NET 面向对象编程语言的特点和结构化异常处理机制，如何利用 Windows 窗体和 Web 窗体及各种控件开发用户界面，如何利用 ADO.NET 访问各种数据源，以及如何利用 ASP.NET 技术开发 Web 应用程序；在附录部分，还专门介绍了如何将用 Visual Basic 以前版本开发的应用程序升级到 Visual Basic .NET。

- 《Visual C++ .NET 程序设计》

作者在介绍 C++ 编程特点时，重点介绍了托管代码的扩展方面，如值类型、结构异常处理、委托和事件的关系，如何使用 .NET 框架提供的各种功能开发托管代码，同时也介绍了如何访问 COM 对象和 Win32 API。

- 《ASP.NET 程序设计》

重点描述了如何建立 Web 应用和 XML Web 服务。本书的作者 G. Andrew Duthie 是资深的网络开发技术方面的专家，长期从事 ASP 和 ASP.NET 有关的培训和咨询工作，具有丰富的建立 Web 应用的经验。本书首先介绍了建立 Web 应用的基础知识，以大量的实例和练习，帮助读者迅速掌握如何通过创建和应用 Web 窗体和服务器控件，建立 Web 应用和 XML Web 服务。对于有志从事开发 Web 应用和 XML Web 服务的人士，本书是一本很好的入门教材。

- 《ADO.NET 程序设计》

ADO.NET 是微软 .NET 框架中的数据访问组件，是一种新的数据访问对象模型。本书重点描述了如何访问各种数据源和建立数据集成系统。具体内容还包括：如何在 Windows 窗体和 Web 窗体中，通过 ADO.NET 绑定数据，如何与 XML 交互，以及如何访问早期的 ADO。

- 《Visual Basic .NET 和 Visual C# .NET 面向对象程序设计》

通过 Microsoft .NET 框架中的两个功能最为强大的编程语言：Visual Basic .NET 和 Visual C# .NET，全面阐述面向对象编程(OOP)思想，并借助具体示例和代码来说明面向对象编程的各种概念。

- 《Web 数据库程序设计(.NET 版)》

基础开发类书籍，介绍了如何借助 .NET 技术，构建更智能化、更高效的 Web 数据库。

ASP.NET 和 ADO.NET 都是当前最新的技术。Visual Basic 和 Visual C++ 也是全球最受欢迎的编程语言，拥有最大的用户群，《Visual Basic .NET 程序设计》和《Visual C++ .NET 程序设计》侧重介绍了这两种编程语言，以及如何用它们构建各种 Windows 桌面应用程序和 Web 应用程序。全套书的

写作遵循了循序渐进的原则，从最新技术的基本概念介绍出发，配有大量的实例和练习，逐步引导读者从一个程序设计门外汉变成程序设计高手；可读性很强，是学习掌握 Visual Studio .NET 的最佳教材。

作为 Visual Studio .NET 产品组的开发人员，我们欣喜地看到，《微软.NET 程序设计系列》将.NET 技术更快、更广地带到每个人的身边。同时也衷心希望，所有读者都能从本丛书中充分享受到.NET 的无穷魅力！

微软中国研究开发中心

Visual Studio .NET 产品组

2002 年 6 月

# 前 言

一直以来，Microsoft Visual Basic 开发人员都强烈要求完全的面向对象语言支持。现在，Microsoft Visual Basic .NET 就可以满足开发人员的这一要求，它提供了面向对象语言的全部特性。此外，整个 Microsoft .NET 框架是依据面向对象的思想设计的，对 Microsoft Windows 应用程序、Web 应用程序、网络服务、图形以及数据访问等都提供开发支持。毋庸置疑，精通面向对象思想的开发人员将成为最优秀的开发者。

对开发人员来说，Microsoft .NET 框架中全新的编程语言还有 C#。这是一个基于 C 语言的编程语言，开发者可以使用 C# 在 .NET 框架下进行开发。而且，有些用 C、Java 和 C++ 开发的项目会转移到 C# 下面，以便利用 .NET 的特性。正在学习 C 语言的 Visual Basic 程序员也可能会转而学习 C#。Visual Basic .NET 和 C# 都支持在 .NET 框架下的面向对象开发。不论选择什么语言进行开发，读懂任何一种语言编写的程序，都会倍增您的可用资源，如 Visual Studio 文档、.NET 书籍、杂志文章和其他开发资源。

## 系 统 要 求

要完成本书中的练习，需要满足下面的硬件和软件要求：

- Microsoft Visual Studio .NET Professional。本书不包括 Visual Studio .NET 软件。在完成本书中的练习之前，必须单独购买软件并安装。
- 一台能够运行 Visual Studio .NET 的计算机。

Visual Studio .NET 网站上推荐如下的硬件配置，详情请访问 <http://msdn.microsoft.com/vstudio/nextgen/>。

- ◆ 计算机处理器  
奔腾 II 级别的处理器，450 MHz；奔腾 III 级别的处理器，推荐 600 MHz。
- ◆ 操作系统  
Microsoft Windows 2000，Server 版或者 Professional 版  
Microsoft Windows XP Home 版或者 Professional 版  
Microsoft Windows NT 4.0 Server
- ◆ 内存  
Windows 2000 Professional，96 MB RAM；推荐 128 MB  
Windows 2000 Server，192 MB；推荐 256 MB



Windows XP Professional, 128 MB; 推荐 160 MB

- ◆ 硬盘  
系统盘需要 500 MB, 安装盘需要 3.0 GB
- ◆ 驱动器  
CD-ROM 驱动器
- ◆ 显示器  
VGA 或者更高分辨率的显示器
- ◆ 输入设备  
Microsoft 鼠标或者兼容的指示设备

## 寻找自己的最佳起点

本书主要介绍面向对象编程的基础知识。对于已经掌握了 Visual Basic 6、Visual Basic .NET、Visual C#或其他 Windows 编程语言基础知识的读者, 都可以使用本书。本书中的练习假设读者已经能够进行下面的操作:

- 新建 Windows 应用程序项目, 并可以生成项目, 运行程序。
- 向 Windows 窗体中添加 Windows 窗体控件。
- 为 Button 控件的 Click 事件创建一个事件处理程序。
- 创建一个简单的方法(在 Visual Basic .NET 中称作 Sub 或 Function)。
- 声明并使用变量。

要了解 Visual Basic .NET, 请参阅 Michael Halvorson 所著的《Visual Basic .NET 程序设计》(清华大学出版社, 2002 年 7 月出版)。要了解 Visual C#, 请参阅孙永强等编著的《Visual C# .NET 程序设计基础》(清华大学出版社, 2002 年 5 月出版); 或参阅 Tom Archer 编著的《C#技术内幕》(清华大学出版社, 2002 年 1 月出版)。

使用下表来寻找自己的最佳起点。

如果您是	按照下面的步骤
刚开始接触面向对象程序设计	按照下一节“安装并使用练习文件”中的指导, 安装本书中的练习。 按照章节的顺序由前至后循序学习面向对象程序设计。第1章~第7章, 第9章以及第11章集中讲述了面向对象程序设计的基本概念, 其他章则讲述了更深层次的概念。

续表

如果您是	按照下面的步骤
从 Visual Basic 6 升级	按照下一节“安装并使用练习文件”中的指导，安装本书中的练习。 按照章节的顺序由前至后循序学习面向对象程序设计。第1章~第7章，第9章以及第11章集中讲述了面向对象程序设计的基本概念，其他章则讲述了更深层次的概念。
从其他面向对象编程语言转过来的	按照下一节“安装并使用练习文件”中的指导，安装本书中的练习。 学习第1章中简述的内容，学会属性和方法的基本语法。 阅读每章结尾的“本章小结”，学习指定的类结构信息。
在完成练习之后，需要参考书中的内容	用目录在书中查找特定主题的内容。 阅读每章结尾的“本章小结”，学习该章中所讲述的语法和技术的简要介绍。

## 安装和使用练习文件

为了帮助读者节省学习时间，作者提供了本书中所有操作需要用到的练习文件。在这些练习文件的帮助下，读者可以集中精力学习如何使用 Visual Basic 和 Visual C#.NET 来进行面向对象编程。通过书中的指导，自己动手操作，学用并举，这显然是获取新知识最简单有效的途径，能达到事半功倍的学习效果。请访问 <http://www.wenyuan.com.cn/> 下载资源，找到本丛书的相关链接，下载打包的练习文件。



### 注意

适合与本书配套使用的软件是 Microsoft Visual Studio .NET Professional，请在学习之前确保安装了适当版本的 Visual Studio。要了解版本信息，请查看软件的产品包装，或启动 Visual Studio，然后在【帮助】菜单中选择【关于 Microsoft Development Environment】，从中查找相关信息。

## 安装练习文件

请按以下步骤安装练习文件：

1. 用 WinZip 或其他解压缩软件对下载的练习文件进行解压缩。
2. 双击 StartCD.exe 文件。

这时将会出现一个启动界面，通过单击界面链接，您可以浏览文件夹内容、安装练习文件，或是访问微软出版社的支持站点。

3. 单击【安装练习文件】链接。  
弹出【安装程序】窗口，提供了可供选择的安装选项。建议接受已经被选中的默认选项。
4. 文件安装结束后，硬盘上会新建一个名为 OOP with Microsoft Visual Basic .NET and Microsoft Visual C# .NET Step by Step 的文件夹，所有的练习文件都放在这个文件夹中。

## 使用练习文件

本书中每一个练习都讲述了什么时候使用练习文件以及怎样使用练习文件。这些文件包含了本书创建的应用程序中所有的源代码和资源，如位图和数据库。读者需要使用它们来完成练习。以下是练习文件中 Visual Basic 项目和 Visual C# 项目的列表：

项 目	描 述
第1章	
ReadBooks	此程序示范了怎样创建类、实例化类，以及怎样使用类。
第2章	
ReadMoreBooks	此程序扩展了第1章中的 ReadBooks 程序，加入了构造函数。
第3章	
CodeAnalysis	这两个应用程序示范了使用类属性的两种方法，以及类属性和 DataGrid 控件之间交互作用的两种方法。
CodeAnalysis2	
第4章	
DeckOfCards	此应用程序通过动态地创建 Windows 窗体控件和拖放操作示范了类方法的用法。
第5章	
TheBank	此程序示范了使用继承的基本操作。
ARoundButton	此小程序示范了如何从一个 Windows 窗体控件派生，并通过重新定义绘图方法来创建一个新的控件。
第6章	
ABetterBank	此程序改写了第5章中的 TheBank 程序，使用抽象类作为基类。
ABetterLibrary	此程序改进了第1章中的 ReadBooks 程序，改用一个强类型的集合类。
Variations	此程序中的几个代码片断示范了其他形式的继承。

项 目	描 述
第7章	
TrainGame	此程序用一个小游戏介绍了委托、事件以及用户自定义的控件。
ThrowSystemException	此小程序中抛出了一个系统异常,然后通过异常处理机制将这个异常捕获。
PersonList	此程序创建并抛出一个自定义的应用程序异常。
第8章	
GamesLibrary	GamesLibrary 项目创建一个组件库,其中包含用于开发简单的
Memory	Memory 纸牌游戏的对象。
第9章	
MoveIt	此应用程序演示了创建和实现接口的基本操作。
Points	Points 项目包含实现 IComparable、Iformattable 和 Ienumerable 接口的对象。
第10章	
PatternMaker	PatternMaker 程序超越了继承的基本用法,它扩展了继承和多态性的用法。
第11章	
BetterCard	此程序改进了第4章中所创建的 Card 类,使用静态方法来消除项目对文件位置的依赖性。
SortablePoint	此应用程序是在第10章中创建的。通过静态方法使该程序更为灵活。
Singleton	静态字段,用于实现 Singleton (单件)设计模式。
第12章	
VectorAlgebra	矢量的数学含义是展示 Visual C# 中加载的运算符的定义和用法。
第13章	
Serialize	Serialize 应用程序展示数据的二进制和 XML 序列化。
DataSetExercise	这是一个非常简单的 ADO.NET 应用程序,它从 Access 数据库读取数据。
第14章	
PatternMaker	此示例使用第10章中的 PatternMaker 练习,展示在程序的最初开发之后修改设计的方法。

## 卸载练习文件

如果使用的操作系统是 Microsoft Windows XP,请按照以下步骤从计算机上卸载练习文件。如果

使用的是其他版本的 Windows 操作系统，请参考 Windows 帮助文档来卸载程序。

1. 单击【开始】|【控制面板】。
2. 在【控制面板】中，单击【添加/删除程序】。
3. 在【添加/删除程序】窗口中，单击选中【目前安装的程序】列表中的 OOP Visual Basic C# .NET Code。
4. 单击【更改/删除】。弹出【确认文件删除】对话框。
5. 单击【是】，练习文件被删除。
6. 删除完成后，关闭【添加/删除程序】窗口。
7. 关闭【控制面板】窗口。

## 本书中的约定和特色段落

本书在版式上力求紧凑美观、便于阅读与学习。书中采用了一些统一的格式规范，以及几种特色段落，提供更简便快捷的学习指南。

### 约定

书中使用了一些特殊的文本格式，从而有助于更加快捷地找到所需信息：

- 每一个练习中都有一系列的任务。每一个任务都有若干步骤。如果某个任务中只有一个步骤，那么这一步前面用实心圆点(●)表示。
- 每一个练习中中文屏幕项(如菜单名、对话框名、标题名及按钮名)均用黑体方括号括起来，以与其他文字区分，如“【文件】菜单”、“【确定】按钮”等等。
- 为了表达更清晰，使用了简写的菜单命令。如“选择【文件】|【新建】|【项目】”，表示首先单击【文件】菜单，指向【新建】，然后单击【项目】。
- 键名之间的加号(+)表示同时按下这些键。如“使用组合键 Alt+Tab”表示同时按下 Alt 键和 Tab 键。
- 键名之间的逗号(,)表示依次按下这些键。如“使用组合键 Alt, F4”表示依次按下 Alt 键和 F4 键。
- 此外，本书使用一些特殊的字体格式来组织内容。下表描述了本书中使用的字体格式。


字体格式	表示	例子
黑体	首次出现的术语、关键词汇	向量是一条具有方向和长度的线段
加粗字体	需要读者输入的内容	输入 <b>XvectorA</b>


续表


字体格式	表示	例子
Courier New字体	键入的程序代码	<pre> \ Visual Basic Public Class Book End Class // Visual C# public class Book { } </pre>
正体英文	布尔值	True, true, False, false
	类名	Book, Library, Train
	控件类型	ListBox, TextBox
	数据类型	String, string, Integer, int
	事件	Click
	窗体	Form1
	命名空间	ReadBooks
	参数类型	String, string, Integer, int
	属性	Name

## 特色段落

书中有两类特色段落，一类带有图标，有如下几种：

 **提示** 对可能的结果或替代方法进行说明。

 **要点** 提供完成一项任务所需的重要信息。

 **注意** 提供与所讨论主题相关的其他信息。

没有图标的楷体段落提供了正文之外更深层次的信息。可能会包含调试信息，设计提示信息或者是读者可能希望进一步了解的主题。

本书每章都以“本章小结”结束，这一节简要回顾了本章所介绍的语法与技巧。

## 勘误、建议和帮助

为了保证本书内容及附加练习文件的准确性，微软出版社通过多种渠道提供支持。请到微软支持网站查询勘误和附加信息，网址为：<http://mspress.microsoft.com/support/>。

如果读者对本书内容或附加练习有疑问、意见或建议，请通过下面的方法反馈到微软出版社：

电子邮件:

mspinput@microsoft.com

邮政地址:

Microsoft Press

Attn: Step by Step Series Editor

One Microsoft Way

Redmond, WA 98052-6399

此外, 也可向本书中文版编辑室发送电子邮件, 以反馈意见。邮件地址请见本书封底上的“读者服务邮箱”。

请注意, 以上地址并不提供对 Visual Studio .NET 软件的产品支持。若要获取有关产品支持的帮助信息, 请访问: <http://support.microsoft.com>。

## 微软出版社网站

微软出版社网站的网址为: <http://www.microsoft.com/mspress>。

微软出版社在网站上提供的信息有: 微软出版社书籍的完整列表、订购信息、相关事件, 以及图书的附加内容等。

此外, 微软公司还在网站上提供了关于 Visual Studio .NET 软件的最新信息, 网址为:

<http://msdn.microsoft.com/vstudio/nextgen/>。

本书由孙瑛霖、丁建扬翻译, 特请 Visual Studio .NET 产品组审校。谨此致谢。正是在他们的大力协助之下, 本书才得以出版。但是, 鉴于我们水平有限, 错误之处在所难免, 敬请读者批评指正。

# 目 录

前言 .....	ix	3.3.3 测试程序 .....	58
<b>第 1 章 编写第一个面向对象程序</b> .....	1	3.4 本章小结 .....	58
1.1 阅读器：您的第一个面向对象程序 .....	1	<b>第 4 章 使用方法</b> .....	61
1.1.1 设计类 .....	2	4.1 纸牌 .....	61
1.1.2 创建 Book 类 .....	4	4.1.1 创建 Card 类 .....	63
1.1.3 在应用程序中使用 Book 类 .....	10	4.1.2 创建 Hand 类 .....	66
1.2 使用类视图 .....	14	4.1.3 测试 Hand 类 .....	72
1.3 创建 ReadBooks 程序 .....	19	4.1.4 创建 Deck 类 .....	75
1.3.1 创建用户界面 .....	19	4.1.5 编写测试程序 .....	81
1.3.2 添加代码 .....	19	4.2 本章小结 .....	89
1.4 本章小结 .....	22	<b>第 5 章 用继承来创建专用类</b> .....	92
<b>第 2 章 用构造函数创建类实例</b> .....	24	5.1 继承简介 .....	92
2.1 构造函数 .....	24	5.2 BankAccount 示例 .....	93
2.2 阅读器的另一种实现方案 .....	25	5.2.1 创建基类 BankAccount 类 .....	94
2.2.1 在 Book 类中创建构造函数 .....	25	5.2.2 创建 SavingsAccount 派生类 .....	97
2.2.2 使用构造函数 .....	29	5.2.3 创建 CheckingAccount 派生类 .....	104
2.3 引用类型和值类型 .....	32	5.2.4 多态地使用派生类 .....	107
2.4 了解数组的声明 .....	34	5.3 从控件继承：RoundButton 类 .....	112
2.5 本章小结 .....	34	5.3.1 创建 RoundButton 类 .....	112
<b>第 3 章 创建字段和属性</b> .....	36	5.3.2 重写 OnPaint 方法 .....	113
3.1 属性设计考虑事项 .....	37	5.3.3 使用类 .....	114
3.2 代码分析器 .....	37	5.4 本章小结 .....	116
3.2.1 设计 SourceFile 类 .....	38	<b>第 6 章 将基类设计为抽象类</b> .....	120
3.2.2 创建 SourceFile 类 .....	39	6.1 抽象类 .....	120
3.2.3 向用户界面添加代码 .....	46	6.2 重新创建 BankAccount 类 .....	121
3.2.4 测试程序 .....	49	6.2.1 用统一建模语言来描述 类的设计 .....	121
3.3 代码分析器的另一种实现 .....	49		
3.3.1 创建 AClass 类 .....	50		
3.3.2 创建用户界面 .....	55		



6.2.2	创建抽象类 .....	123	8.3.3	编写游戏运行过程代码 .....	204
6.2.3	编写 SavingsAccount 类 .....	125	8.4	本章小结 .....	205
6.2.4	编写 CheckingAccount 类 .....	128	<b>第 9 章</b>	<b>通过接口提供服务 .....</b>	<b>207</b>
6.2.5	测试类 .....	131	9.1	IMoveable 接口 .....	207
6.3	类型化集合类 .....	132	9.1.1	定义 IMoveable 接口 .....	208
6.3.1	重新设计 Library 类 .....	133	9.1.2	在 Pawn 类中实现 IMoveable 接口 .....	210
6.3.2	创建类 .....	134	9.1.3	测试 IMoveable 接口 .....	213
6.3.3	重新创建 Library 类 .....	137	9.2	.NET 框架中的接口 .....	214
6.3.4	测试类 .....	138	9.2.1	实现 IComparable 接口 .....	215
6.4	本章小结 .....	141	9.2.2	实现 IEnumerable 接口和 IEnumerator 接口 .....	219
<b>第 7 章</b>	<b>用事件和异常响应更改 .....</b>	<b>143</b>	9.3	接口的另外两个用途 .....	231
7.1	事件驱动应用程序示例: 火轨游戏 .....	143	9.4	本章小结 .....	232
7.1.1	实现 Track 类 .....	145	<b>第 10 章</b>	<b>通过多态实现类的互换 .....</b>	<b>234</b>
7.1.2	创建 CaughtOnFire 事件 .....	150	10.1	图案制作程序 .....	234
7.1.3	实现 Train 类 .....	155	10.2	图案制作程序的设计 .....	235
7.1.4	实现用户界面 .....	158	10.3	基类 .....	237
7.2	不用设计器设置事件方法 .....	161	10.3.1	创建 Pattern 类 .....	237
7.3	异常 .....	164	10.3.2	创建 PatternEditor 类 .....	238
7.3.1	产生异常 .....	164	10.4	派生类 .....	239
7.3.2	编写异常类 .....	165	10.4.1	创建 DrawnPattern 类 .....	239
7.4	本章小结 .....	170	10.4.2	创建 DrawnPatternEditor 类 .....	242
<b>第 8 章</b>	<b>用组件构造程序 .....</b>	<b>173</b>	10.4.3	创建 BitmapPattern 类 .....	246
8.1	Memory 游戏 .....	173	10.4.4	创建 BitmapPatternEditor 类 .....	247
8.2	游戏类库 .....	175	10.5	用户界面 .....	250
8.2.1	创建 GamesLibrary 项目 .....	175	10.5.1	创建用户界面元素 .....	251
8.2.2	更改命名空间 .....	175	10.5.2	创建模板实例 .....	253
8.2.3	创建 Card 控件 .....	176	10.5.3	编辑和保存新图案 .....	254
8.2.4	创建 Memory 游戏项目 .....	183	10.5.4	测试应用程序 .....	256
8.2.5	创建 Deck 组件 .....	185	10.6	本章小结 .....	256
8.3	Memory 游戏应用程序 .....	203			
8.3.1	将 LotsOfFun.Games 控件 添加到工具箱 .....	203			
8.3.2	创建用户界面 .....	204			