

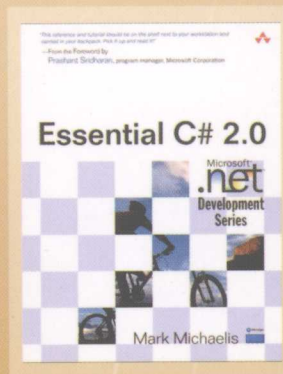
Essential C# 2.0

C#本质论

[美] Mark Michaelis 著
周靖 译



- Amazon全五星图书
- 在与其他语言的比较中轻松学习
- 微软C#产品经理强烈推荐



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

.NET系列

TP312/2903

2008

Essential C# 2.0

C#本质论

[美] Mark Michaelis 著

周靖 译

人民邮电出版社
北 京

图书在版编目 (CIP) 数据

C# 本质论 / (美) 米凯利斯 (Michaelis, M.) 著; 周靖译. —北京: 人民邮电出版社, 2008.7
(图灵程序设计丛书)
书名原文: Essential C# 2.0
ISBN 978-7-115-18187-9

I. C… II. ①米…②周… III. C 语言—程序设计—教材
IV. TP312

中国版本图书馆CIP数据核字 (2008) 第075260号

内 容 提 要

本书是一本清晰、简明的 C# 教程, 涵盖了 C# 2.0。书中对 C# 语言的每个重要结构都用简短的示例代码进行说明, 并且和其他语言进行了全方位比较。每章开头的“思维导图”指明了本章要讨论的主题以及每个主题同整体的关系。全书由 18 章和 3 个附录组成。在简单介绍了 C# 之后, 重点讨论了 C# 的数据类型、运算符、方法、类等基本概念, 随后还对泛型、迭代器、反射、线程、互操作性等高级主题进行了深入而透彻的讨论。

本书适用于对 C# 感兴趣的各種层次的读者, 无论是初学者还是高级编程人员, 都能从本书中获益。

图灵程序设计丛书

C#本质论

-
- ◆ 著 [美] Mark Michaelis
 - 译 周 靖
 - 责任编辑 刘艳娟
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 32.5
字数: 870千字 2008年7月第1版
印数: 1—4 000册 2008年7月河北第1次印刷

著作权合同登记号 图字: 01-2007-0866号

ISBN 978-7-115-18187-9/TP

定价: 69.00元

读者服务热线: (010)88593802 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版权声明

Authorized translation from the English language edition, entitled: *Essential C# 2.0*, 0321150775 by Mark Michaelis, published by Pearson Education, Inc., publishing as Addison-Wesley Professional. Copyright © 2007 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD. and Posts & Telecommunications Press Copyright © 2008.

本书中文简体字版由Pearson Education Asia Ltd.授权人民邮电出版社独家出版。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。

本书封面贴有Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

版权所有，侵权必究。

对本书的赞誉

“本书完成了一项非常艰巨的任务。前面的章节很易于被刚入门的开发者理解，而在后面的章节中，作者将自己对于语言的理解倾囊以授，并为有经验的开发者提供了发挥C# 2.0最大潜力所需的详细信息。Mark是组织内容的高手。从第1章起Mark就成功赢得了许多高手的心。与此同时，全书的所有内容都恰如其分。”

——Chris Kinsman, Vertafore首席架构师, Microsoft社区区域经理

“真是令人耳目一新！这本书透彻地探讨了C#，而不是像某些书那样，只对整个.NET框架走马观花地介绍一下就完事。本书对新手和专家都极其有价值。”

——Jon Skeet, 微软C# MVP

“本书是有经验的程序员掌握当今最热门的C#语言的一站式参考书。Mark以一种易于理解的方式讲解了C#的方方面面。初学者能通过本书打下一个良好的基础，并在此基础上构建其下一代的应用程序。”

——Stephen Toub, 技术编辑, MSDN Magazine

“本书全面讲述了C#的所有编程构架。通过恰当地组织，编程新手能通过本书快速入门，并引导更多有经验的编程人员进入结构化编程世界。由于本书一直都将重点放在C#的基本编程构造上，比如泛型、委托等，所以本书是不可或缺的。另外，对于想用C#的最新特性来解决其日常编程问题的程序员，本书也是不可或缺的。”

——Narendra Poflee, IT集成专家, Itron Inc.

“本书对于所有对C#有兴趣的程序员来说都是一本理想图书。假如你是一名新手，可以通过本书快速掌握基本的C#编程知识，并逐渐熟悉大量基本概念。本书的学习流程设计得非常贴心，已经讲过的概念不会反复讲解。对于有经验的程序员，本书也具有非常高的含金量。强烈推荐已经熟悉C#的程序员阅读本书。无论如何，这会成为我案头必备的参考书。”

——Michael Stokesbary, 软件工程师, Itron Inc.

术有专攻，日益精进

——译者序

最近在看《史记 孔子世家》，撷取孔子学琴一则与各位共勉：

孔子学鼓琴师襄子，十日不进。师襄子曰：“可以益矣。”孔子曰：“丘已习其曲矣，未得其数也。”有间，曰：“已习其数，可以益矣。”孔子曰：“丘未得其志也。”有间，曰：“已习其志，可以益矣。”孔子曰：“丘未得其为人也。”有间，有所穆然深思焉，有所怡然高望而远志焉。曰：“丘得其为人，黯然而黑，几然而长，眼如望羊，如王四国，非文王其谁能为此也！”师襄子辟席再拜，曰：“师盖云文王操也。”

这则故事的大意为孔子向师襄子学琴，学了十天并不要求学习新的东西。尽管师襄子一再敦促可以增加新的内容，但孔子坚信，不达到三个境界，不能继续学习新的东西：学习掌握演奏的技巧，领会其中的志趣，熟悉乐曲的作者。最终，孔子领悟到乐曲作者的心境，使师襄子叹服不已。

每学习一种新东西，都离不开这三个境界，正所谓“昨夜西风凋碧树，独上高楼，望尽天涯路”，此境界一也；“衣带渐宽终不悔，为伊消得人憔悴”，此境界二也；“众里寻他千百度，回头蓦见，那人正在，灯火阑珊处”，此境界三也。

从事技术图书的翻译多年，经常有读者来信问：“我想学习编程，如何入手？”对待这种问题，我往往都会勉励他们，万丈高楼平地起，首先要搭建好整个学习框架，梳理整个学习层次，然后选择合适的入门书夯实基础，完全掌握之后再按照已经搭建好的框架逐一填充各个学习模块。

作为一种功能强大、简单好用的语言，C#主要面向需要使用Microsoft .NET Framework来创建应用程序的开发人员。它在C++和Visual Basic的基础上去芜存菁，最终被打造为一种更清晰、逻辑性更强的语言。它可以用于开发Windows窗体应用程序、访问Microsoft SQL Server数据库、开发ASP.NET Web应用程序以及创建并使用Web服务。C# 2.0新增了一些重要的特性，如泛型、迭代器和匿名方法等。Visual Studio提供的开发环境使这些特性易于使用，而它新增的大量向导和增强措施能显著提高开发人员的工作效率。在效率为王的今天，C#无疑是很多开发人员的首选编程语言。

本书是一本清晰、简明的C#教程，无论初学者还是专家，都能从中获得有益的内容。本书对C#和其他语言进行了全方位的比较。书中全面讨论了C#语言，每个重要的结构都用简短的示例代码来进行说明。完整的示例代码可从网上获取。Mark Michaelis对本书的内容进行了精心组织，读者很容易融会贯通。每章开头的“思维导图”指明了本章要讨论的主题以及每个主题同整体的关系。

无论你是否是一名刚开始接触编程的新手，是打算了解C#的一名有经验的开发者，还是一名合格的C#程序员，现在只是想了解一下C# 2.0的新特性，都能从本书中获得需要的内容，快速上手并开始编写

C#应用程序。

在本书翻译过程中，我被作者深厚的技术功底和毫不保留的精神所折服。C#虽然比较容易上手，但它的部分主题对于初学者来说仍然是颇为深奥的。但是，在作者的笔下，一个复杂的主题总能被讲解得简单易懂，而且深入透彻。仅仅这一点，就使市面上的大多数同类书籍黯然失色。

在此，我要感谢原书作者Mark Michaelis，他是一位非常具有激情和活力的技术专家。翻译过程中，他热情、耐心地解释我所提出的问题，并虚心、坦诚地采纳了我所提出的修改意见。另外，还要感谢我的家人，尤其是女儿周子衿，这个仅有9岁的小姑娘，会经常从一些新奇的角度来帮助我重新认识这个世界。最后，感谢各位读者对我的支持和信赖，我会尽自己的微薄之力，继续恪尽职守，为大家奉献更好的译作。

衷心祝愿读者朋友能通过本书，开始愉快而激动人心的C#之旅！

周 靖

2007年于北京

序

多年来，C语言不断吸收新的概念，发展出新的语言，从面向对象编程(C++)到跨平台运行(Java)，然后是构成C#基础的高度结构化模型。对于我们这个行业中目睹了这一演变的人来说，很显然，C#已经万事俱备，完全有资格成为一种占据优势地位的开发语言。

本书提供了开发者快速上手和实际运用C#语言而需知道的一切。许多简洁的代码示例、用于提供上下文的思维导图以及Mark将C#同其他语言对比时采用的有趣方式，使本书在众多的C#图书中脱颖而出！

我认识Mark已经有不少日子了。他广泛参与了C#的开发过程，加上他之前有C++、COM+和C#图书的写作经验，因此我相信他足以写出一部影响深远的C#著作。

除此之外，Mark的教学经验使他能够将专业技能与实际的、直观的教学结合起来。虽然本书是从C#对象编程的原理讲起的，但它也可以作为在复杂系统中查证高级多线程技术的参考。无论是新手还是专家，都会对Mark的行文风格以及内容组织感到由衷的欣赏。

我坚信这本参考书兼教程会成为你的良师益友，现在就开始阅读吧！我完全相信，无论是C#语言的新手，还是有经验的开发者，都会在阅读中体验到“灵机一动”的感觉！

Prashant Sridharan
时任微软C#产品经理
2006年3月

前言

在软件工程的发展历史中，用于编写计算机程序的方法经历了几次思维模式的重大转变。每一种思维模式都是以前一种思维模式为基础的，其宗旨都是增强代码的组织，并降低复杂性。学习本书的过程，就是思维模式不断发生改变的过程。

本书开始的几章指导你学习**顺序编程结构**（sequential programming structure）。在这种模型中，语句是按照执行顺序来写的。这种模型的问题在于，随着需求的增加，复杂性也将呈指数级地增加。为了降低复杂性，代码块被转变成方法，从而产生了**结构化编程模型**（structured programming model）。在这种模型中，你可以从一个程序中的多个位置调用同一个代码块，而不必在程序中重复这些代码。然而，即使有这种结构，程序还是会很快变得臃肿不堪，需要进行进一步的抽象。所以，在此基础上，人们又提出了面向对象编程的概念，这将在第5章进行讨论。在后续的几章中，你将继续学习其他方法，比如基于接口的编程，并最终学习通过attribute进行初级的声明式编程（第14章）。

本书有以下3个主要职能。

- ❑ 全面讲述C#语言，其内容已经远远超越了一本简单的教程，为你进行高效率软件开发打下坚实的基础。
- ❑ 对于已经熟悉了C#的读者，本书提供了一些更加复杂的编程思想，并深入讨论了在语言的新版本（C# 2.0）中引入的特性。
- ❑ 它是你永远的案头参考，即便在你精通了这种语言之后。

成功学习C#的关键在于，要尽可能快地开始编程。不要等自己成为一名理论上的“专家”之后，才开始写代码。所以，不要犹豫，马上开始写你的程序吧。作为迭代开发^①的追随者，我希望即使是一名刚开始学习编程的新手，也能在学到本书第2章末尾的时候，动手开始写基本的C#代码。

有许多主题都没有在本书中进行讨论。你在本书找不到ASP.NET、ADO.NET、智能客户端开发以及分布式编程等主题。虽然这些主题与.NET Framework有关，但它们都值得用专门的、分专题进行讲述。幸运的是，已经有丰富的图书供读者选择。不过，在读完本书之后，你在上述任何领域继续深入学习都会有游刃有余的感觉。本书的重点在于C#以及基类库中的类型。

① 简单地说，迭代开发是指分周期、分阶段进行一个项目，以增量方式逐渐对其进行改进的过程。——译者注

本书面向的读者

写作本书时，我面临的一个挑战在于，如何在持续吸引高级开发人员眼球的同时，不用使用类似“assembly”“link”、“chain”、“thread”和“fusion”的字眼儿来打击初学者的信心，否则许多人会认为这是一本冶金方面的书，而不是讲程序设计的。^①本书的主要读者是已经有一定编程经验，并想多学一种语言来“防身”的开发者。然而，我小心地制定了本书的内容配置，使其对于所有级别的开发者来说，都有足够大的价值。

- ❑ 初学者：假如你是一名编程新手，本书将帮助你从入门级的程序员过渡成为一名C#开发者，消除对以后摆在你面前的任何C#编程任务的害怕心理。本书不仅要教会你语法，还要教你养成良好的编程习惯，为你将来的编程生涯打下良好的基础。
- ❑ 熟悉结构化编程的程序员：学习外语最好的方法就是“沉浸法”^②。类似地，学习一门计算机语言最好的方法就是在动手中学习，而不是等熟知了它的所有“理论”之后再动手。基于这个前提，本书最开始的内容是那些熟悉结构化编程的开发者很容易上手的。到第4章结束时，这些开发者应该可以开始写基本的控制程序。然而，要成为一名真正的C#开发者，记住语法只是第一步。为了从简单程序过渡到企业级开发，C#开发者必须基于对象及其关系来思考问题。为此，第5章的“初学者主题”开始介绍类和面向对象开发。对于C、COBOL和FORTRAN等结构化编程语言来说，虽然它们仍在发挥作用，但作用会越来越小。所以，软件工程师们应该逐渐开始了解面向对象开发。C#是进行这一思维模式转变的理想语言，它本来就是基于“面向对象开发”这一中心思想来设计的。
- ❑ 熟悉“基于对象”和“面向对象”理念的开发者：C++和Java程序员以及许多有经验的Visual Basic程序员都可归于此类。对于分号和大括号，他们可是一点儿都不陌生！简单浏览一下第1章的代码，你会发现，在其核心，C#类似于你熟知的C和C++风格的语言。
- ❑ C#专家：对于已经精通C#的人，本书可供你参考不太常见的语法。此外，对于其他地方强调较少的语言细节以及微妙之处，作者提出了自己的见解。最重要的是，本书提供了编写可靠和易于维护的代码的指导原则及模式。你教别人学习C#时，本书也颇有助益。

C# 2.0问世后，一些最重要的增强包括：

- ❑ 分部类（参见第5章）；
- ❑ 全局命名空间限定符global：（参见第9章）；
- ❑ 为属性的getter和setter使用的不同访问修饰符（参见第5章）；
- ❑ 匿名方法（参见第13章）；
- ❑ 泛型（参见第11章）；
- ❑ 迭代器主题（参见第12章）。

考虑到许多人还不熟悉这些主题，本书围绕它们进行了详细的讨论。涉及高级C#开发的还有“指

① 上述每个单词在计算机和冶金领域都有专门的含义，所以作者用它们开了一个玩笑。例如，assembly既是“程序集”，也是“装配件”；thread既是“线程”，也是“螺纹”。——译者注

② 沉浸法，即immersion approach，是指想办法让学习者泡到一个全外语的环境中，比如孤身一人在国外生活或学习。——译者注

针”这一主题，该主题将在第17章讨论。即使一些有经验的C#开发者，也经常无法很好地理解这一主题。

本书特色

本书是一本遵循核心C# Language 2.0 Specification的语言参考书。为了帮助读者理解各种C#结构，书中用大量例子演示了每一种特性，而且为每个概念都提供了相应的指导原则和最佳实践，确保代码能顺利编译，避免留下隐患，并获得最佳的可维护性。

为了增强可读性，所有代码均进行了特殊格式处理，而且每一章的内容都使用思维导图来概括。

代码示例

本书大多数代码片段都能在“公共语言基础结构”(Common Language Infrastructure, CLI)的任何实现上运行，其中包括Mono、Rotor和Microsoft .NET平台。我很少使用平台或厂商特有的库，除非需要解释只和那些平台有关的某些重要概念(例如，解释如何正确处理Windows单线程用户界面)。C# 2.0特有的主题在附录C进行了总结。

下面是一个示例代码清单。

代码清单1-17 在代码中添加注释

```
class CommentSamples
{
    static void Main()
    {
        单行注释
        string firstName; // Variable for storing the first name
        string lastName; // Variable for storing the last name

        System.Console.WriteLine("Hey you!");

        语句内部带分隔符的注释
        System.Console.Write /* No new line */ (
            "Enter your first name: ");
        firstName = System.Console.ReadLine();

        System.Console.Write /* No new line */ (
            "Enter your last name: ");
        lastName = System.Console.ReadLine();

        带分隔符的注释
        /* Display a greeting to the console using composite formatting. */
        System.Console.WriteLine("Your full name is {0} {1}.",
            firstName, lastName);
        // ...
    }
}
```

下面简单介绍一下我采用的代码格式。

- 注释以斜体显示。

```
/* Display a greeting to the console  
using composite formatting */
```

- 关键字加粗显示。

```
static void Main()
```

- 有的代码被突出显示，是为了指明这些代码与早先列出的有所区别，或是为了演示正文中介绍的概念。

```
System.Console.Write /* No new line */ (
```

突出显示的内容可能是一整行，也可能仅仅是一行中的几个字符。

```
System.Console.WriteLine(  
    "Your full name is {0} {1}.",
```

- 不完整的程序清单包含一个省略号，指出不相关的代码已被省略。

```
// ...
```

- 在代码清单之后，列出了对应的控制台输出，如下所示：

输出1-4

```
>HeyYou.exe  
Hey you!  
Enter your first name: Inigo  
Enter your last name: Montoya
```

用户在执行程序时输入的内容显示为斜体。

虽然我可以提供完整的代码示例，让你复制到自己的程序中，但这样做会使你从一个特定的主题上分心。因此，你需要自行修改代码示例，然后才能把它们集成到自己的程序中。此外，书中的代码示例没有显式地包含using System语句。在所有的例子中，这个语句都是必需的。

可以从以下网址下载示例代码：

<http://mark.michaelis.net/EssentialCSharp>

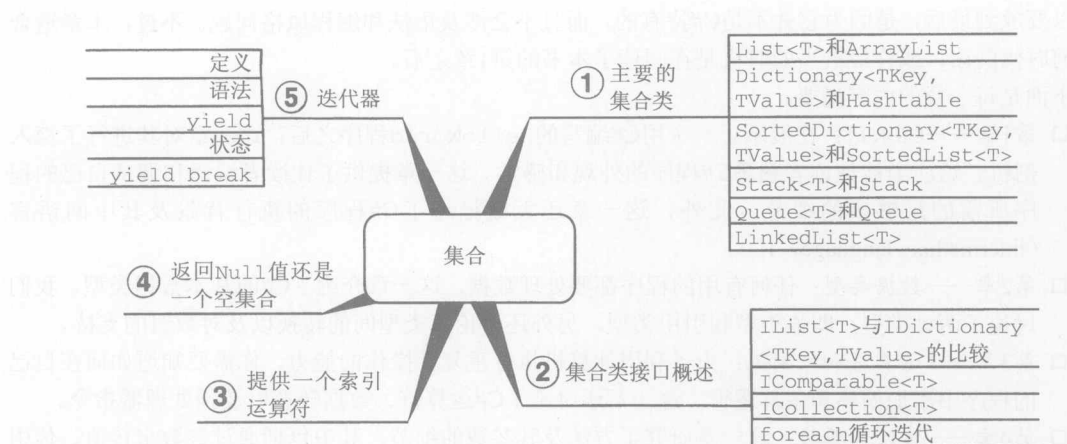
或者

www.awprofessional.com/title=0321150775

思维导图

每一章开头都包含一幅“思维导图^①”。作为一个提纲，它的作用是为读者提供对每章内容的一个快速参考。下面是一个例子（摘自第12章）。

① 思维导图，即mind map，又称脑图、心智图，其作用是帮助你学习、组织和存储你想要的所有信息，它以自然的方式对信息进行分类，使你能够立即得到你想要的一切。你可以将其想象成一幅帮助自己记忆和思考的思维路线图。——译者注



每一章的主题显示在思维导图的中心，高级主题围绕这个中心展开。利用思维导图，读者可以方便地搭建自己的知识体系，可以从一个主题出发，更清楚地理解围绕它的各个具体概念，避免中途纠缠于一些不相干的枝节问题。

有用的注释

根据自己的编程水平，利用书中特殊的代码块和注释，你可以轻松地定位自己要阅读的内容。

- 初学者主题：特别针对入门级程序员提供的定义或解释。
- 高级主题：可以让有经验的开发者将注意力放在他们最关心的内容上。
- 标注：用标注框^①来强调关键原则，使读者对其重要性一目了然。
- 语言对比：分散在正文中的补充内容，描述了C#和其他语言的关键差异，为熟悉其他语言的读者提供指引。

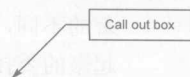
本书的内容组织

站在一个较高的角度来看，软件工程的宗旨就是对复杂性进行管理。本书正是基于这个宗旨来组织内容的。我的目的很简单，就是降低你学习编程的复杂性。第1章～第4章介绍的是结构化编程，通过这些内容的学习，你可以立即开始写一些功能简单的代码。第5章～第9章介绍了C#的面向对象构造。新手应在完全理解了这几章的内容之后，再开始接触本书其余部分的更高级主题。

第11章～第13章介绍了更多用于降低复杂性的构造，讲解了当今几乎所有程序都要用到的通用模式。在理解了它们之后，你可以更加轻松地理解如何通过反射和attribute来实现动态编程。第14章～第17章将广泛运用反射和attribute来实现线程处理和互操作性。

本书最后专门用一章（第18章）来讲述CLI。这一章在开发平台的背景下对C#语言进行了描述。

^① 标注框，即callout box，是一种需要准确排版的书稿元素。下面展示了一个典型的callout box。



之所以要放到最后，是因为它并不是C#特有的，而且不会涉及语法和编程风格问题。不过，本章适合在任何时候阅读，或许最恰当的时机是在阅读了本书的第1章之后。

下面是每一章的内容提要。

- 第1章——C#入门：在展示了一个用C#编写的HelloWorld程序之后，这一章对其进行了深入剖析。这应当能使读者熟悉C#程序的外观和感觉。这一章提供了让读者编译和调试自己的程序所需的足够多的细节。此外，这一章还大概描述了C#程序的执行背景及其中间语言（intermediate language, IL）。
- 第2章——数据类型：任何有用的程序都要处理数据，这一章介绍了C#的基本数据类型。我们讨论了两种类型，即值类型和引用类型，另外还讨论了类型间的转换以及对数组的支持。
- 第3章——运算符和控制流：为了利用计算机执行重复性操作的能力，你需要知道如何在自己的程序中添加循环和条件逻辑。这一章还讨论了C#运算符、数据转换以及预处理器指令。
- 第4章——方法和参数：这一章研究了方法及其参数的细节，其中包括通过参数来传值、传引用和返回数据。C#不支持默认参数，这一章解释了这背后的原因，并讨论了相同的功能如何实现。
- 第5章——类：前面已经学过了类的基本构成元素，这一章将这些构造合并到一起，从而获得具有完整功能的类型。类是面向对象技术的核心，它为一“类”对象定义了一个模板。
- 第6章——继承：虽然继承是许多开发者的基本编程手段，但C#提供了一些独一无二的构造，比如new修饰符。这一章讨论了继承语法的细节，其中包括重写。
- 第7章——接口：这一章展示了如何利用接口来定义类之间的“可以进行版本控制的交互契约”（versionable interaction contract）。C#同时包含显式和隐式的接口成员实现，可以实现一个额外的封装等级，这是其他大多数语言所不支持的。
- 第8章——值类型：尽管不如定义引用类型那么流行，但在有些情况下仍然有必要定义其类似于C#内置基本类型的值类型。这一章要介绍如何定义结构，同时揭示它们的特性。
- 第9章——合式类型：这一章讨论了一些更高级的类型定义，解释了如何实现运算符，比如“+”和转型运算符，并描述了如何将多个类封装到一个库中。除此之外，这一章还演示了如何定义命名空间和XML注释，并讨论了如何基于垃圾回收机制来设计令人满意的类。
- 第10章——异常处理：这一章是对第4章引入的异常处理机制的一个延伸讨论，描述了如何利用异常层次结构来创建自定义异常。另外，还强调了异常处理的一些最佳实践。
- 第11章——泛型：从某种意义上说，泛型或许是C#1.0缺少的一个最重要的特性。这一章全面讨论了这个新特性。
- 第12章——集合：有了泛型之后，.NET 1.1中的所有集合类都可以替换成它们的泛型等价物。这一章讨论了这些集合类，并描述了对其共同行为进行定义的接口。由于迭代器对底层“运行时”的影响是微乎其微的，C# 2.0通过迭代器简化了集合的创建。迭代器用一个更清晰的语法来指定如何遍历类中的数据。
- 第13章——委托和事件：正是因为有了委托，才使C#与其前身语言（C和C++等）相比有了显著的不同，它定义了代码中处理事件的模式。这几乎完全杜绝了写轮询例程的必要。封装起来的委托（称为事件）是公共语言运行时（Common Language Runtime, CLR）的一个核心构造。另外，这一章还探讨了匿名方法，它也是C# 2.0的一个新特性。

- ❑ 第14章——反射和attribute: 20世纪80年代末, 程序结构的思维模式发生了根本性的变化, 面向对象的编程是这个变化的基础。类似地, attribute使声明性编程和嵌入式元数据成为可能, 因而引入了一种新的思维模式。这一章探讨了attribute的方方面面, 并讨论了如何通过反射机制来获取它们。这一章还讨论了如何通过基类库(Base Class Library, BCL)中的序列化框架来实现文件的输入和输出。
- ❑ 第15章——多线程处理: 大多数现代的程序都要求使用线程来执行长时间运行的任务, 同时确保对并发的进行快速响应。随着程序变得越来越复杂, 必须采取额外的措施在这些高级环境中保护数据。多线程应用程序的编写是一项复杂的任务。这一章讨论了如何操作线程, 并讲述了如何采取一些必要的措施来防止将多线程应用程序弄得一团糟。
- ❑ 第16章——多线程处理模式: 这一章以第15章为基础, 演示了如何利用一些内建的线程处理模式来简化对多线程代码的显式控制。
- ❑ 第17章——平台互操作性和不安全的代码: 必须认识到的是, C#是相对年轻的一种语言, 有许多现有的代码是用其他语言写成的。为了用好这些现有的代码, C#通过P/Invoke提供了对非托管代码的支持。除此之外, C#允许使用指针, 也允许执行直接内存操作。虽然含有指针的代码要求特殊的权限才能运行, 但它具有与C风格的API完全兼容的能力。
- ❑ 第18章——CLI: 事实上, C#被设计成在CLI的顶部工作的一种最有效的编程语言。这一章讨论了C#程序与底层“运行时”及其规范的关系。
- ❑ 附录A——下载和安装C#编译器与CLI平台: 这个附录介绍了如何安装Microsoft .NET和Mono, 它们是编译和运行C#程序的基础平台。
- ❑ 附录B——完整源代码清单: 本书许多章都将源代码打散到多个代码清单中。如果代码清单较大, 就会使读者难以跟进。这个附录将第3章、第11章、第12章、第14章和第17章的代码清单整合成完整的程序, 便于读者从整体上理解各个单独的代码清单。
- ❑ 附录C——C# 2.0主题: 提供了所有C# 2.0主题的快速参考, 旨在帮助C# 1.0程序员快速熟悉2.0的新特性。

希望本书成为你学习和掌握C#技能的一个好帮手。另外, 希望以后需要了解C#的一些特殊主题及其内部工作原理的时候本书也是一本出色的参考书。

致谢

世界上没有任何一本书是作者单枪匹马就能出版的, 在此, 我要向此过程中帮助过我的所有人致以衷心的感谢。

表达感激之情的顺序并不重要, 我是想到谁就感谢谁。到现在为止, 为了让我顺利完成此书, 我的家人做出了巨大的牺牲。在Banjamin、Hanna和Abigail眼中, 他们的爸爸经常因为此书而分心, 但Elisabeth承受的更多。家里的大事小事全靠她一个人, 她独自承担家庭的重任。亲爱的, 辛苦了!

为保证本书技术上的准确性, 许多技术编辑对本书中的各章都进行了仔细审阅。我常常惊讶于他们的认真程度, 任何不易察觉的小错误都逃不过他们的火眼金睛, 他们是Paul Bramsman、Doug Dechow、Gerard Frantz、Anson Horton、Angelika Langer、Nicholas Paldino、Narendra Poflee、Jon Skeet、Michael Stokesbary、John Timney和Stephen Toub。

特别要感谢Michael, 在我编写本书期间, 他在编辑技术内容和完善本书结构方面, 给予我莫大的帮助, 更不用说他给予我的友情, 如同珍宝。同时, 我还要特别感谢两位C# MVP (微软最有价值专家) Nicholas和John, 他们对于特定领域内这一编程语言的了解仅次于C#开发团队。

此外, 还要感谢Addison-Wesley的所有员工, 感谢他们在与我合作期间所表现出来的耐心, 容忍我将注意力频频转移到书稿之外的其他方面。谨向以下人员致以衷心的感谢: Sheri Cain、Jessica D'Amico、Curt Johnson、Joan Murray、Stephane Nakib和Ann Wells。

尤其要感谢Ann, 从本书提出选题开始, 她就一直指导和鼓励我。

最后, 感谢微软公司开发部的Prashant Sridharan。是他使我开始动笔写作本书, 而且他提供的大量素材使我难以置信地占尽了先机。

目 录

第1章 C# 概述.....1	2.3 null 和 void.....31
1.1 Hello, World.....1	2.3.1 null.....31
1.1.1 应用程序的编译和运行.....2	2.3.2 void.....32
1.1.2 托管执行和公共语言基础结构.....3	2.4 类型的分类.....32
1.2 C#语法基础.....4	2.4.1 值类型.....32
1.2.1 C#关键字.....4	2.4.2 引用类型.....33
1.2.2 类型定义.....6	2.5 可空修饰符.....34
1.2.3 Main.....6	2.6 数据类型之间的转换.....35
1.2.4 语句和语句分隔符.....7	2.6.1 显式转型.....35
1.2.5 空白.....8	2.6.2 隐式转型.....37
1.3 使用变量.....9	2.6.3 不进行转型的类型转换.....38
1.3.1 数据类型.....9	2.7 数组.....39
1.3.2 变量的声明.....10	2.7.1 数组的声明.....39
1.3.3 变量的赋值.....10	2.7.2 数组的实例化和赋值.....40
1.3.4 变量的使用.....11	2.7.3 数组的使用.....43
1.4 控制台输入和输出.....11	2.7.4 字符串作为数组使用.....47
1.4.1 从控制台获取输入.....12	2.7.5 常见错误.....49
1.4.2 将输出写入控制台.....13	2.8 小结.....50
1.5 注释.....14	第3章 运算符和控制流.....51
1.6 CIL 和 ILDASM.....16	3.1 运算符.....51
1.7 小结.....18	3.1.1 一元运算符正和负.....52
第2章 数据类型.....19	3.1.2 二元算术运算符.....52
2.1 基本数值类型.....19	3.1.3 圆括号运算符.....57
2.1.1 整数类型.....19	3.1.4 赋值运算符.....58
2.1.2 浮点类型.....20	3.1.5 递增和递减运算符.....58
2.1.3 decimal 类型.....21	3.1.6 常量表达式.....61
2.1.4 字面值.....21	3.2 流控制概述.....61
2.2 更多基本类型.....24	3.2.1 if 语句.....63
2.2.1 布尔类型.....24	3.2.2 嵌套 if.....64
2.2.2 字符类型.....25	3.3 代码块.....65
2.2.3 字符串.....26	3.4 作用域.....67