Microsoft

9

Ninth Edition

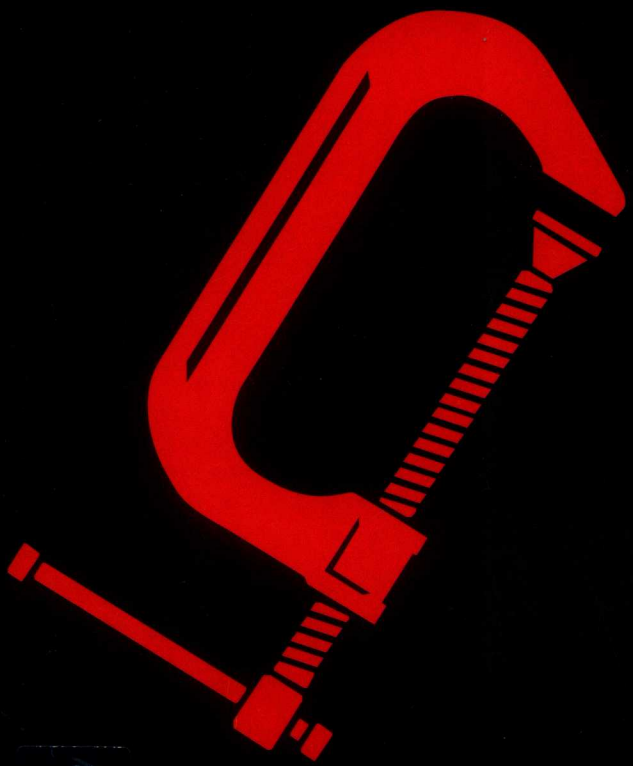
- 承袭Step-by-Step风格，完美演绎C#精髓
- 久经考验、深受国内外读者喜爱的C#入门经典

Visual C#

从入门到精通（第9版）

[英] 约翰·夏普（John Sharp） 著
周 靖 译

Step by step



MVP
Microsoft®
Most Valuable
Professional

清华大学出版社

Visual C#从入门到精通
(第9版)

[英] 约翰·夏普(John Sharp) 著

周 靖 译

清华大学出版社

北 京

内 容 简 介

C#作为微软的旗舰编程语言，深受程序员喜爱，是编写高效应用程序的首选语言。Visual C# 2017 提供了大量新功能，本书围绕语言的基础知识和这些新功能全面介绍了如何利用 Visual Studio 2017 和.NET Framework 4.6.1 编写 C#应用程序。本书沿袭深受读者欢迎的 Step by Step 风格，通过合理的练习引导读者逐步构建在 Windows 10 上运行的应用程序、访问 SQL Server 数据库以及开发多线程应用等。

全书共 27 章，结构清晰，叙述清楚。所有练习均在 Visual Studio 2017 简体中文版上进行过全面演练。无论是刚开始接触面向对象编程的新手，还是打算迁移到 C#的 C、C++或 Java 程序员，都可以从本书汲取到新的知识。迅速掌握 C#编程技术。

Authorized translation from the English language edition, entitled MICROSOFT VISUAL C# STEP BY STEP, Ninth Edition, 9781509307760 by SHARP, JOHN, published with the authorization of Microsoft Corporation by 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 TSINGHUA UNIVERSITY PRESS LIMITED Copyright © 2019.

本书中文简体翻译版由Pearson Education授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号　图字号：01-2018-4943
本书封面贴有清华大学出版社防伪标签，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 C#从入门到精通/(英)约翰·夏普(John Sharp)著；周靖译. —9 版. —北京：清华大学出版社，2019
书名原文：Microsoft Visual C# Step by Step, Ninth Edition
ISBN 978-7-302-51624-8

Ⅰ. ①V…　Ⅱ. ①约…　②周…　Ⅲ. ①C 语言－程序设计　Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 257331 号

责任编辑：文开琪
装帧设计：杨玉兰
责任校对：周剑云
责任印制：丛怀宇
出版发行：清华大学出版社
网　　址：http://www.tup.com.cn, http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　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：三河市铭诚印务有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　张**：42.75　　**字　数**：1026 千字
版　　次：2019 年 1 月第 9 版　　**印　次**：2019 年 1 月第 1 次印刷
定　　价：139.00 元

产品编号：080785-01

前　言

Microsoft Visual C#是一种功能强大、使用简单的语言，主要面向需要使用Microsoft .NET Framework来创建应用程序的开发者。它在C++和Microsoft Visual Basic的基础上去芜存菁，最终形成一种更加清晰、更富有逻辑的语言。

- C# 1.0于2001年亮相。
- 几年后随着C# 2.0和Visual Studio 2005的问世，语言新增了几个重要功能，包括泛型、迭代器和匿名方法等。
- 随同Visual Studio 2008发布的C# 3.0添加了更多功能，包括扩展方法、Lambda表达式以及语言集成查询(Language Integrated Query，LINQ)。
- 2010年发布的C# 4.0继续增强，改善了与其他语言和技术的互操作性。新增功能包括具名参数和可选参数，另外还有`dynamic`类型(指示语言的“运行时”对对象进行晚期绑定)。在随C# 4.0发布的.NET Framework中，最重要的新功能就是“任务并行库”(Task Parallel Library，TPL)。可用TPL构建具有良好伸缩性的应用程序，从而快速和简单地发挥出多核处理器的潜力。
- C# 5.0通过`async`方法修饰符和`await`操作符提供了对异步任务的原生支持。
- C# 6.0是一次增量式升级，提供了许多有利于简化开发的功能，包括字符串插值(再也不需要`String.Format`了)，改进的属性实现方式，表达式主体方法等。
- C# 7.0进一步增强，提高了生产力并移除了C#一些不合时宜的设计。例如，现在属性访问器方法可作为表达式主体成员实现，方法支持以元组形式返回多个值，简化了`out`参数的用法，`switch`语句开始支持模式和类型匹配。还有其他许多更新，本书将一一阐述。[①]

虽然Windows 10是运行C#应用程序最重要的平台，但现在也可通过.NET Core运行时在其他操作系统(包括 Linux)上运行用 C#写的代码。这使程序更容易在多种环境中运行。另外，Windows 10支持高度交互性的应用程序，它们可以进行数据共享和协作，还可以连接云服务。Windows 10最引人注目的是对UWP(Universal Windows Platform，通用Windows平台)应用的支持。这种应用设计在任何Windows 10设备上运行，无论是全功能的桌面系统、笔记本和平板，还是资源有限的智能手机和物联网设备。熟悉C#的核心功能后，下一步就是掌握如何开发能在所有这些平台上运行的应用。

① 译注：访问本书中文博客（*http://bookzhou.com*）了解和上一版的区别。

语音激活是另一个值得关注的功能。Windows 10 提供了 Cortana(小娜)个人语音数字助理。可将自己的应用和 Cortana 集成来实现数据搜索和其他操作。虽然自然语音分析天生就很复杂，但让应用响应 Cortana 的请求却令人惊讶地简单。详情将在第 26 章描述。另外，从大规模企业应用程序到手机上运行的移动应用，云已成为许多系统架构的重要元素，本书最后一章会讲解如何开发基于云的应用。

Visual Studio 2017 开发环境使这些强大功能变得容易使用，大量新向导和增强显著提高了开发效率。写作这本书给我带来了许多乐趣，希望你的阅读亦是如此！

下面列出和本书《Visual C#从入门到精通》第 8 版相比的主要变化(从译者的角度整理)。

- 2.3.3 节“指定数值”，强调了显式指定值的类型的重要性。
- 3.1.5 节“从方法返回多个值”，引入元组概念，从一个方法返回多个值。这章开始引入表达式主体方法，注意虽然都用=>操作符，但表达式主体方法和 Lambda 表达式有本质的不同。相当于=>的两个重载。
- 3.3.2 节“嵌套方法”，本节提到的 Factorial 解决方案在学生文件中不可用。解决方案是用空白的 DailyRate 解决方案。
- 6.2.4 节“筛选异常”，`catch (...) when (...) {}`。
- 6.4.1 节“使用 throw 表达式”，用?:操作符简化抛出异常的代码。例如：`string name = nameField.Text != "" ? nameField.Text : throw new Exception("未输入值");`。不这样写，就要写一长串 if...else 语句。
- 7.4.3 节“解构对象”，讲的是和元组配合使用。Deconstruct 方法和 out 参数相配合，向元组中的变量赋值。不要把解构器和析构器弄混了。
- 8.2.1 节“空条件操作符”，在对象上调用其方法时，用?.操作符判断对象是否为 null。例如，`A?.B?.C?.Do(E);` 。其中，ABC 任何一个求值为 null，E 都不会执行(短路)。
- 8.8.3 节“复习 switch 语句”用 switch 代替一系列 `if (expr is type varname)... else if(expr is type varname)...`语句，从而简化空引用检查(不用 is 操作符)。case 还能加 when 表达式来进一步限制条件。
- 10.1.11 节“访问包含值类型的数组”，讲的是一些传统数组处理方法，在数组元素从引用类型修改成值类型后会出错，因为现在返回拷贝而非引用。解决方案是用 ref 关键字返回引用。例如，`ref type method(...){... return ref 数组元素}`。注意两个地方添加了 ref。调用也要改，变成 `ref type variable = ref method(...)`。
- 第 15 章的简单属性大量改为使用表达式主体方法。例如，`get { return this._x; }`改为 `get => this._x;`。

此为试读，需要完整PDF请访问：www.ertongbook.com

- 第 16 章介绍了新的常量表达式，可以直接将二进制赋给变量了：`uint binData = 0b01111;`。`0b0` 是二进制，`0x0` 是十六进制。眼睛看不过来还可以加下划线(编译器会忽略)，例如 `uint moreHexData = 0x0_F0_5A_CC_0F;`。
- 21.2.6 节的练习和原书不符，解决方案并没有一开始就集成 BinaryTree 项目，需自己添加。
- 24.1.6 节“任务、内存分配和效率”解释了如何用 Cache-Aside 设计模式把异步方法设计成大多数时候都同步执行，耗时的、经常重复的计算的结果放到缓存中。需要用 NuGet 包管理器下载 System.Threading.Tasks.Extensions 包。
- 第 27 章对有关 Azure 云的内容进行了全面修订，代码简化了不少。

本书面向的读者

本书假定你要使用 Visual Studio 2017 和.NET Framework 4.6.1 学习基础的 C#编程知识。学完本书后，会对 C#有一个全面和透彻的理解，会用它开发响应灵敏、易于伸缩的 Windows 10 应用程序。

本书不面向的读者

本书面向刚开始用 C#进行开发的人士，重点放在 C#语言上面。本书不涉及企业级 Windows 应用程序的开发技术，比如 ADO.NET、ASP.NET、Windows Communication Foundation 或者 Workflow Foundation。要了解这些方面的知识，可参考微软出版社的其他书籍。

本书的组织

全书分为四部分。

- 第 I 部分“Visual C#和 Visual Studio 2017 概述”介绍 C#语言的核心语法，还演示了 Visual Studio 编程环境。
- 第 II 部分“理解 C#对象模型”深入探讨如何用 C#创建和管理新类型，如何管理这些类型引用的资源。
- 第 III 部分“用 C#定义可扩展类型”全面讨论如何利用 C#语言元素来构建能在多个应用程序中重用的类型。
- 第 IV 部分“用 C#构建 UWP 应用”描述通用 Windows 10 编程模型，以及如何用 C#为新模型构建交互式应用程序。

导读

本书帮助你掌握多个基本领域的开发技能。无论刚开始学习编程，还是从另一种语言(C、C++、Java 或 Visual Basic)转向 C#，本书都能提供帮助。参考下表找到最佳起点。

读者类型	步骤
面向对象编程的新手	1. 按照“示例代码”一节的步骤安装练习文件 2. 顺序阅读第 I 部分、第 II 部分和第III部分 3. 有一定经验后，如有兴趣，继续完成第Ⅳ部分的学习
熟悉 C 语言等过程编程语言，但新涉足 C#	1. 按照“示例代码”一节的步骤安装练习文件 2. 略读前 5 章来获得对 C#和 Visual Studio 2017 的大致印象，重点阅读第 6 章～第 22 章 3. 阅读第Ⅳ部分了解如何构建可伸缩的 Windows 应用
从面向对象语言 C++或 Java 等迁移到 C#	1. 按照“示例代码”一节的步骤安装练习文件 2. 略读前 7 章，获得对 C#语言和 Visual Studio 2017 的大致印象，重点阅读第 8 章～22 章 3. 阅读第Ⅳ部分了解如何构建 UWP 应用
从 Visual Basic 迁移到 C#	1. 按照“示例代码”一节的步骤安装练习文件 2. 顺序阅读第 I 部分、第 II 部分和第III部分 3. UWP 应用的开发请阅读第Ⅳ部分 4. 阅读每章末尾的“快速参考”小节，了解 C#和 Visual Studio 2017 特有的构造
做完所有练习后再将本书用作为参考书	1. 按目录查主题 2. 阅读章末“快速参考”，查看语法和技术要点归纳

本书大多数章节都通过实例方便读者巩固刚学到的知识。无论感兴趣的是哪个主题，都注意先下载并安装好示例代码。

本书的约定和特色

本书通过一些约定来增强内容的可读性，以便于读者理解。

- 每个练习都用编号的操作步骤来完成。
- “注意”等特色段落提供了成功完成一个步骤需要了解的额外信息或替代方案。
- 要求读者输入的文本**加粗**显示。
- 两个键名之间的加号(+)意味着必须同时按下这两个键。例如，“按组合键 Alt+

Tab”意味着按住 Alt 键，再按 Tab 键。

- 描述菜单操作时，采用“文件”|“打开”的形式，意思是从“文件”菜单中选择“打开”命令。

系统需求

为了完成本书的练习，需准备以下硬件和软件：

- Windows 10 家庭、专业、教育或企业版，版本 1507 或以上
- Visual Studio 2017 社区、专业或企业版的最新版本。安装时最起码选择以下工作负载：
  - 通用 Windows 平台开发
  - .NET 桌面开发
  - ASP.NET 和 Web 开发
  - Azure 开发
  - 数据存储和处理
  - .NET Core 跨平台开发
- 1.8 GHz 或更快的处理器(推荐双核或以上)
- 2 GB RAM(推荐 4GB，在虚拟机中运行再加 512 MB)
- 10 GB 可用硬盘空间
- 1024 × 768 或更高分辨率显卡
- 下载软件和示例代码需要 Internet 连接

取决于 Windows 配置，可能需要以管理员身份安装和配置 Visual Studio 2017。

电脑上要启用开发人员模式以创建和运行 UWP 应用。详情参考“启用设备进行开发”(*https://msdn.microsoft.com/library/windows/apps/dn706236.aspx*)。

示例代码

本书大多数章节都包含互动练习供练手。从以下网址下载示例代码(包括练习完成前后两种格式)：

http://aka.ms/VisCSharp9e/downloads

http://bookzhou.com

安装示例代码很简单，将下载的 zip 文件解压到“文档”文件夹即可。[①]

使用示例代码

本书每一章都解释了在什么时候以及如何使用练习文件。需要练习文件时，书中会给出相应的指示，帮助你打开正确文件。

重要提示 许多例子都依赖示例代码没有包含的 NuGet 包。这些包在首次生成项目时自动下载(有的需手动搜索并安装)。如首次打开一个项目且不生成，Visual Studio 可能报告大量引用无法解析的错误。生成一次即可完成引用的解析，错误将消失。

下表总结了本书用到的所有 Visual Studio 2017 项目和解决方案，它们以文件夹的形式进行分组以便查找。练习通常会为同一个项目提供初始文件和完成之后的版本。已完成的项目存储在带有- Complete 后缀的文件夹中。

项目/解决方案	说明
第 1 章	
TextHello	作为第一个项目，它指导你创建一个简单程序来显示欢迎文本
Hello	打开一个窗口，提示用户输入姓名并显示个性化的欢迎辞
第 2 章	
PrimitiveDataTypes	演示如何使用基元类型声明变量，如何向变量赋值，如何在窗口中显示值
MathsOperators	演示算术操作符(+、-、*、/、%)
第 3 章	
Methods	改进上个项目的代码，体会如何使用方法来建立代码的结构
DailyRate	指导你写自己的方法，执行方法，使用 Visual Studio 2017 调试器来单步执行方法
DailyRate Using Optional Parameters	演示如何让方法获取可选参数，如何使用具名参数来调用方法
第 4 章	
Selection	演示如何用嵌套 `if` 语句实现复杂逻辑，例如比较两个日期的相等性
SwitchStatement	这个简单的程序使用一个 `switch` 语句将字符转换成相应的 XML 形式
第 5 章	
WhileStatement	用 `while` 语句逐行读取源文件，在窗体上的文本框中显示每一行
DoStatement	使用 `do` 语句将十进制数转换成八进制

① 译注：本书将路径“C:\Users*YourName*\Documents”简称为“文档”文件夹。可在 Windows 文件资源管理器的地址栏输入环境变量**%UserProfile%\Documents** 打开该文件夹。

续表

项目/解决方案	说明
第 6 章	
MathsOperators	改进第 2 章的 MathsOperators 项目，试验会造成程序执行失败的各种未处理异常。然后用 `try` 和 `catch` 关键字使应用程序更健壮，防止因为错误输入或操作而失败
第 7 章	
Classes	演示如何定义自己的类，为它添加公共构造器、方法和私有字段；还演示如何用 `new` 关键字创建类的实例，如何定义静态方法和字段
第 8 章	
Parameters	演示值类型和引用类型的参数的区别，还演示如何使用 `ref` 和 `out` 关键字
第 9 章	
StructsAndEnums	定义结构来表示日期
第 10 章	
Cards	使用数组建模纸牌游戏中的一手牌
第 11 章	
ParamsArrays	演示如何使用 `params` 关键字使方法能接受任意数量的实参
第 12 章	
Vehicles	用继承创建简单交通工具类，还演示如何定义虚方法
ExtensionMethod	演示如何为 `int` 类型创建扩展方法，允许将整数从十进制转换成其他进制
第 13 章	
Drawing	实现图形绘图包的一部分。用接口定义要由几何图形对象公开并实现的方法
Drawing Using Interfaces	扩展 Drawing 项目，将几何图形对象的常用功能集成到抽象类中
第 14 章	
GarbageCollectionDemo	演示如何使用 Dispose 模式实现异常安全的资源清理
第 15 章	
Drawing Using Properties	扩展第 13 章的 Drawing 项目，用属性封装类的数据
AutomaticProperties	演示如何为类创建自动属性，如何用它们初始化类的实例
第 16 章	
Indexers	该项目使用了两个索引器，一个根据姓名查找电话号码，另一个根据电话号码查找姓名
第 17 章	
BinaryTree	演示如何使用泛型生成类型安全的结构，可包含任何类型的元素
BuildTree	演示如何使用泛型实现类型安全的方法，可获取任何类型的参数
第 18 章	
Cards	升级第 10 章的代码，演示如何用集合建模一手牌

续表

项目/解决方案	说明
第 19 章	
BinaryTree	演示如何实现泛型 IEnumerator<T>接口，为泛型 Tree 类创建枚举器
IteratorBinaryTree	用迭代器为泛型 Tree 类生成枚举器
第 20 章	
Delegates	演示如何通过委托调用方法，将方法的逻辑和调用方法的应用程序分开。然后对项目进行扩展，演示如何用事件提醒对象发生了某事，以及如何捕捉事件并执行必要的处理
第 21 章	
QueryBinaryTree	演示如何通过 LINQ 查询从二叉树对象获取数据
第 22 章	
ComplexNumbers	定义新类型来建模复数，并为这种类型实现常用的操作符
第 23 章	
GraphDemo	生成并在 UWP 窗体上显示复杂图表。用单线程执行计算
Parallel GraphDemo	使用 Parallel 类对创建和管理任务的过程进行抽象
GraphDemo With Cancellation	中途得体地取消任务
ParallelLoop	演示何时不该使用 Parallel 类创建和运行任务
第 24 章	
GraphDemo	修改第 23 章的同名项目，使用 async 关键字和 await 操作符异步计算图表数据
PLINQ	使用并行任务，用 PLINQ 查询数据
CalculatePI	使用统计学采样计算 PI 的近似值。使用了并行任务
第 25 章	
Customers	实现能自动适应不同屏幕分辨率和设备大小的 UI。UI 使用 XAML 样式更改字体和背景图片
第 26 章	
DataBinding	修改上一章的 Customers 项目，使用数据绑定在 UI 中显示从数据源获取的客户资料；还演示了如何实现 INotifyPropertyChanged 接口，从而允许 UI 更新客户资料，并将改动发送回数据源
ViewModel	这个版本的 Customers 项目通过实现 Model-View-ViewModel 模式，将 UI 同数据源访问逻辑分开
Cortana	将 Customers 应用和 Cortana 集成。用户可发出语音指令来搜索客户
第 27 章	
Web Service	该解决方案包含一个 Web 应用程序来提供 ASP.NET Web API Web 服务，使 Customers 应用能从 SQL Server 数据库获取客户数据。Web 服务通过由实体框架创建的实体模型来访问数据库

此为试读，需要完整PDF请访问：www.ertongbook.com

简明目录

第 I 部分　Visual C#和 Visual Studio 2017 概述

第 1 章　欢迎进入 C#编程世界 3
第 2 章　使用变量、操作符和表达式 27
第 3 章　方法和作用域 49
第 4 章　使用判断语句 78
第 5 章　使用复合赋值和循环语句 95
第 6 章　管理错误和异常 111

第 II 部分　理解 C#对象模型

第 7 章　创建并管理类和对象 135
第 8 章　理解值和引用 155
第 9 章　使用枚举和结构创建值类型 177
第 10 章　使用数组 195
第 11 章　理解参数数组 217
第 12 章　使用继承 228
第 13 章　创建接口和定义抽象类 249
第 14 章　使用垃圾回收和资源管理 273

第 III 部分　用 C#定义可扩展类型

第 15 章　实现属性以访问字段 295
第 16 章　处理二进制数据和使用索引器 317
第 17 章　泛型概述 333
第 18 章　使用集合 360
第 19 章　枚举集合 381
第 20 章　分离应用程序逻辑并处理事件 394
第 21 章　使用查询表达式来查询内存中的数据 421
第 22 章　操作符重载 443

第 IV 部分　用 C#构建 UWP 应用

第 23 章　使用任务提高吞吐量 463
第 24 章　通过异步操作提高响应速度 500
第 25 章　实现 UWP 应用的用户界面 538
第 26 章　在 UWP 应用中显示和搜索数据 578
第 27 章　在 UWP 应用中访问远程数据库 618
译者后记 662

目　录

第 I 部分　Visual C#和 Visual Studio 2017 概述

第 1 章　欢迎进入 C#编程世界 3

1.1　开始在 Visual Studio 2017 环境中编程 3

1.2　写第一个程序 6

1.3　使用命名空间 11

1.4　创建图形应用程序 13

1.4.1　探索通用 Windows 平台应用程序 20

1.4.2　向图形应用程序添加代码 23

1.4.3　其他类型的图形应用程序 25

小结 26

第 1 章快速参考 26

第 2 章　使用变量、操作符和表达式 27

2.1　理解语句 27

2.2　使用标识符 27

认识关键字 28

2.3　使用变量 29

2.3.1　命名变量 29

2.3.2　声明变量 29

2.3.3　指定数值 30

2.4　使用基元数据类型 31

2.4.1　未赋值的局部变量 31

2.4.2　显示基元数据类型的值 32

2.5　使用算术操作符 37

2.5.1　操作符和类型 37

2.5.2　深入了解算术操作符 38

2.5.3　控制优先级 43

2.5.4　使用结合性对表达式进行求值 43

2.5.5　结合性和赋值操作符 44

2.6　变量递增和递减 45

2.7　声明隐式类型的局部变量 46

小结 47

第 2 章快速参考 47

第 3 章　方法和作用域 49

3.1　创建方法 49

3.1.1　声明方法 49

3.1.2　从方法返回数据 50

3.1.3　使用表达式主体方法 51

3.1.4　调用方法 52

3.1.5　从方法返回多个值 54

3.2　使用作用域 57

3.2.1　定义局部作用域 57

3.2.2　定义类作用域 58

3.2.3　重载方法 58

3.3　编写方法 59

3.3.1　重构代码 63

3.3.2　嵌套方法 66

3.4　使用可选参数和具名参数 69

3.4.1　定义可选参数 70

3.4.2　传递具名参数 70

3.4.3　消除可选参数和具名参数的歧义 71

小结 75

第 3 章快速参考 76

第 4 章　使用判断语句 78

4.1　声明布尔变量 78

4.2　使用布尔操作符 78

4.2.1　理解相等和关系操作符 79

4.2.2　理解条件逻辑操作符 79

4.2.3　短路求值 80

4.2.4　操作符的优先级和结合性总结 80

4.3　使用 if 语句做出判断 81

4.3.1 理解 if 语句的语法 81
4.3.2 使用代码块分组语句 82
4.3.3 嵌套 if 语句 83
4.4 使用 switch 语句 88
4.4.1 理解 switch 语句的语法 89
4.4.2 遵守 switch 语句的规则 90
小结 93
第 4 章快速参考 94

第 5 章 使用复合赋值和循环语句 95

5.1 使用复合赋值操作符 95
5.2 编写 while 语句 96
5.3 编写 for 语句 100
5.4 编写 do 语句 102
小结 109
第 5 章快速参考 109

第 6 章 管理错误和异常 111

6.1 处理错误 111
6.2 尝试执行代码和捕捉异常 111
6.2.1 未处理的异常 113
6.2.2 使用多个 catch 处理程序 114
6.2.3 捕捉多个异常 114
6.2.4 筛选异常 115
6.2.4 传播异常 119
6.3 使用 checked 和 unchecked 整数运算 121
6.3.1 编写 checked 语句 121
6.3.2 编写 checked 表达式 122
6.4 抛出异常 124
6.5 使用 finally 块 129
小结 130
第 6 章快速参考 131

第 II 部分 理解 C#对象模型

第 7 章 创建并管理类和对象 135

7.1 理解分类 135
7.2 封装的目的 135
7.3 定义并使用类 136
7.4 控制可访问性 137
7.4.1 使用构造器 138
7.4.2 重载构造器 139
7.4.3 解构对象 146
7.5 理解静态方法和数据 147
7.5.1 创建共享字段 148
7.5.2 使用 const 关键字创建静态字段 149
7.5.3 理解静态类 149
7.5.4 静态 using 语句 149
7.5.5 匿名类 152
小结 153
第 7 章快速参考 153

第 8 章 理解值和引用 155

8.1 复制值类型的变量和类 155
8.2 理解 null 值和可空类型 160
8.2.1 空条件操作符 161
8.2.2 使用可空类型 162
8.2.3 理解可空类型的属性 163
8.3 使用 ref 和 out 参数 163
8.3.1 创建 ref 参数 164
8.3.2 创建 out 参数 165
8.4 计算机内存的组织方式 167
8.5 System.Object 类 169
8.6 装箱 169
8.7 拆箱 170
8.8 数据的安全转型 171
8.8.1 is 操作符 171
8.8.2 as 操作符 172
8.8.3 复习 switch 语句 172
小结 175
第 8 章快速参考 175

第 9 章 使用枚举和结构创建值类型 177

9.1 使用枚举 177
9.1.1 声明枚举 177
9.1.2 使用枚举 177

9.1.3 选择枚举字面值 178
9.1.4 选择枚举的基础类型 179
9.2 使用结构 181
9.2.1 声明结构 182
9.2.2 理解结构和类的区别 183
9.2.3 声明结构变量 185
9.2.4 理解结构的初始化 185
9.2.5 复制结构变量 189
小结 193
第 9 章快速参考 193

第 10 章 使用数组 195

10.1 声明和创建数组 195
10.1.1 声明数组变量 195
10.1.2 创建数组实例 196
10.1.3 填充和使用数组 196
10.1.4 创建隐式类型的数组 197
10.1.5 访问单独的数组元素 198
10.1.6 遍历数组 199
10.1.7 数组作为方法参数和返回值传递 200
10.1.8 复制数组 201
10.1.9 使用多维数组 202
10.1.10 创建交错数组 203
10.1.11 访问包含值类型的数组 212
小结 215
第 10 章快速参考 215

第 11 章 理解参数数组 217

11.1 回顾重载 217
11.2 使用数组参数 218
11.2.1 声明参数数组 219
11.2.2 使用 params object[] 220
11.2.3 使用参数数组 221
11.3 比较参数数组和可选参数 224
小结 226
第 11 章快速参考 227

第 12 章 使用继承 228

12.1 什么是继承 228
12.2 使用继承 229
12.2.1 复习 System.Object 类 230
12.2.2 调用基类构造器 231
12.2.3 类的赋值 232
12.2.4 声明新方法 233
12.2.5 声明虚方法 234
12.2.6 声明重写方法 235
12.2.7 理解受保护的访问 238
12.3 创建扩展方法 243
小结 247
第 12 章快速参考 247

第 13 章 创建接口和定义抽象类 249

13.1 理解接口 249
13.1.1 定义接口 250
13.1.2 实现接口 250
13.1.3 通过接口引用类 252
13.1.4 使用多个接口 252
13.1.5 显式实现接口 253
13.1.6 接口的限制 254
13.1.7 定义和使用接口 255
13.2 抽象类 263
13.3 密封类 265
13.3.1 密封方法 265
13.3.2 实现并使用抽象类 266
小结 271
第 13 章快速参考 272

第 14 章 使用垃圾回收和资源管理 273

14.1 对象生存期 273
14.1.1 编写析构器 274
14.1.2 为什么要使用垃圾回收器 275
14.1.3 垃圾回收器的工作原理 277
14.1.4 慎用析构器 277
14.2 资源管理 278
14.2.1 资源清理方法 278
14.2.2 异常安全的资源清理 279
14.2.3 using 语句和 IDisposable 接口 279

14.2.4 从析构器中调用 Dispose 方法 281
14.3 实现异常安全的资源清理 283
小结 289
第 14 章快速参考 290

第 III 部分 用 C#定义可扩展类型

第 15 章 实现属性以访问字段 295
15.1 使用方法实现封装 295
15.2 什么是属性 297
15.2.1 使用属性 299
15.2.2 只读属性 300
15.2.3 只写属性 300
15.2.4 属性的可访问性 301
15.3 理解属性的局限性 301
15.4 在接口中声明属性 303
15.5 生成自动属性 308
15.6 用属性初始化对象 310
小结 314
第 15 章快速参考 314
第 16 章 处理二进制数据和使用索引器 317
16.1 什么是索引器 317
16.1.1 存储二进制值 317
16.1.2 显示二进制值 318
16.1.3 操纵二进制值 318
16.1.4 用索引器解决相同问题 320
16.1.5 理解索引器的访问器 322
16.1.6 对比索引器和数组 322
16.2 接口中的索引器 324
16.3 在 Windows 应用程序中使用索引器 325
小结 331
第 16 章快速参考 331
第 17 章 泛型概述 333
17.1 object 的问题 333
17.2 泛型解决方案 336
17.2.1 对比泛型类和常规类 338
17.2.2 泛型和约束 339
17.3 创建泛型类 339
17.3.1 二叉树理论 339
17.3.2 使用泛型构造二叉树类 342
17.4 创建泛型方法 350
17.5 可变性和泛型接口 353
17.5.1 协变接口 354
17.5.2 逆变接口 356
小结 357
第 17 章快速参考 358
第 18 章 使用集合 360
18.1 什么是集合类 360
18.1.1 List<T>集合类 361
18.1.2 LinkedList<T>集合类 363
18.1.3 Queue<T>集合类 365
18.1.4 Stack<T>集合类 366
18.1.5 Dictionary<TKey, TValue>集合类 367
18.1.6 SortedList<TKey, TValue>集合类 368
18.1.7 HashSet<T>集合类 369
18.2 使用集合初始化器 371
18.3 Find 方法、谓词和 Lambda 表达式 372
18.4 比较数组和集合 375
小结 379
第 18 章快速参考 379
第 19 章 枚举集合 381
19.1 枚举集合中的元素 381
19.1.1 手动实现枚举器 382
19.1.2 实现 IEnumerable 接口 386
19.2 用迭代器实现枚举器 388
19.2.1 一个简单的迭代器 388

19.2.2 使用迭代器为 Tree<TItem> 类定义枚举器 390
小结 392
第 19 章快速参考 393

第 20 章 分离应用程序逻辑并处理事件 394

20.1 理解委托 394
20.1.1 .NET Framework 类库的委托例子 395
20.1.2 自动化工厂的例子 397
20.1.3 不用委托实现工厂控制系统 397
20.1.4 用委托实现工厂控制系统 398
20.1.5 声明和使用委托 400
20.2 Lambda 表达式和委托 407
20.3 启用事件通知 408
20.3.1 声明事件 409
20.3.2 订阅事件 410
20.3.3 取消订阅事件 410
20.3.4 引发事件 410
20.4 理解用户界面事件 411
小结 418
第 20 章快速参考 418

第 21 章 使用查询表达式来查询内存中的数据 421

21.1 什么是 LINQ 421
21.2 在 C#应用程序中使用 LINQ 422
21.2.1 选择数据 424
21.2.2 筛选数据 426
21.2.3 排序、分组和聚合数据 427
21.2.4 联接数据 429
21.2.5 使用查询操作符 430
21.2.6 查询 Tree<TItem>对象中的数据 432
21.2.7 LINQ 和推迟求值 437
小结 440
第 21 章快速参考 441

第 22 章 操作符重载 443

22.1 理解操作符 443
22.1.1 操作符的限制 443
22.1.2 重载的操作符 444
22.1.3 创建对称操作符 445
22.2 理解复合赋值 447
22.3 声明递增和递减操作符 447
22.4 比较结构和类中的操作符 448
22.5 定义成对的操作符 449
22.6 实现操作符 449
22.7 理解转换操作符 455
22.7.1 提供内建转换 455
22.7.2 实现用户自定义的转换操作符 456
22.7.3 再论创建对称操作符 457
22.7.4 添加隐式转换操作符 457
小结 459
第 22 章快速参考 460

第 IV 部分 用 C#构建 UWP 应用

第 23 章 使用任务提高吞吐量 463

23.1 使用并行处理执行多任务处理 463
23.2 用.NET Framework 实现多任务处理 464
23.2.1 任务、线程和线程池 465
23.2.2 创建、运行和控制任务 466
23.2.3 使用 Task 类实现并行处理 469
23.2.4 使用 Parallel 类对任务进行抽象 478
23.2.5 什么时候不要使用 Parallel 类 482
23.3 取消任务和处理异常 484
23.3.1 协作式取消的原理 485
23.3.2 为 Canceled 和 Faulted 任务使用延续 496

此为试读，需要完整PDF请访问：www.ertongbook.com