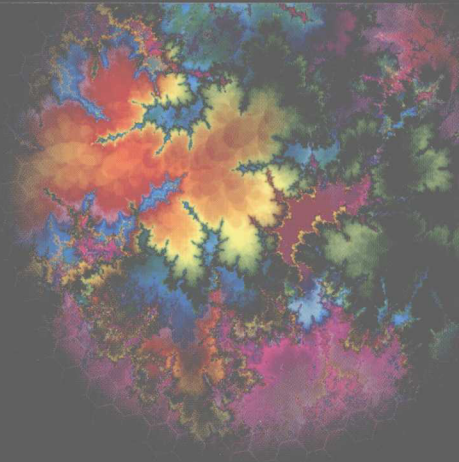


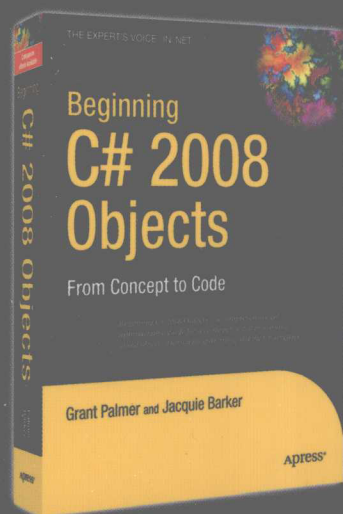
THE EXPERT'S VOICE® IN .NET



Beginning C# 2008 Objects 中文版

——概念到代码 (第2版)

(美) Grant Palmer 著
Jacquie Barker 著
薛莹 任鸿 程文俊 译



Apress®



清华大学出版社

Beginning C# 2008 Objects 中文版

——概念到代码(第2版)

Grant Palmer
(美) Jacquie Barker 著

薛莹 任鸿 程文俊 译

清华大学出版社

北 京

Grant Palmer, Jacquie Barker

Beginning C# 2008 Objects: From Concept to Code

EISBN: 978-1-4302-1088-7

Original English language edition published by Apress, 2855 Telegraph Avenue, #600, Berkeley, CA 94705 USA. Copyright © 2008 by Apress L.P. Simplified Chinese-language edition copyright © 2010 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Apress 出版公司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: 01-2009-0538

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

Beginning C# 2008 Objects 中文版——概念到代码(第2版)/(美)帕默(Palmer, G.), (美)芭克(Barker, J.) 著; 薛莹, 任鸿, 程文俊 译. —北京: 清华大学出版社, 2010.6

书名原文: Beginning C# 2008 Objects: From Concept to Code

ISBN 978-7-302-22347-4

I. B… II. ①帕… ②芭… ③薛… ④任… ⑤程… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2010)第 059248 号

责任编辑: 王 军 李 阳

装帧设计: 孔祥丰

责任校对: 成凤进

责任印制: 杨 艳

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

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

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 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: 37.25 字 数: 860 千字

版 次: 2010 年 6 月第 1 版 印 次: 2010 年 6 月第 1 次印刷

印 数: 1~3000

定 价: 78.00 元

产品编号: 030220-01

亚马逊网友评论精选

★★★★★Best Book to Learn OOP in C#, March 17, 2010

By Frank Stepanski (PA)

Object Oriented Programming (OOP) has been around for 20 years and is the basis for anytime of intermediate to advanced programming in any language. C# is no different and for any beginner who knows the basics of C# language, it is crucial for them to understand how objects work in C#

This book is a great book in taking the somewhat complicated theory and practice of OOP development and explaining it so anybody with some previous C# experience can understand. Of course if you have other programming background and not C#, you still will be able to follow the book, it will just make it a little easier if you know the specific C# syntax.

The authors go through from the beginning in explaining what objects are all about along with how they interact, relationships between objects, how collections work, polymorphism, etc. But also includes some excellent topics that you don't often see in C# books such as object modeling, formalizing requirements with use cases, uml modeling, file persistence, etc.

It really goes through the entire gambit of OO development from beginning to end. The code examples that are explained throughout the book are great and really help cement what the authors are trying to get across to the reader.

Code examples, diagrams, notes and lesson review material all make it easy for the reader to get to what they want to learn without going through endless pages of theory. Each topic is very well explained and can be used as a learning tool or as a reference book.

I highly recommend this book to any programmer who wants to take their C# skills to the next level.

面向对象编程(OOP)概念的出现大约已有20年了,无论使用哪种语言编写中高级程序,OOP总是起到基础性的作用。C#语言与其他语言一样,对于了解C#语言基础的每一位初学者,理解对象在C#中的工作原理至关重要。

本书是一本相当有价值的书籍,清楚地解释了OOP开发中较复杂的理论和实践,任何具有使用C#经验的开发人员都能够理解它们。当然,如果您具有其他语言(而不是C#语言)的编程背景,您仍然能够顺利学习本书,因为本书对C#语法的介绍非常简单明了。

本书首先解释了对象的含义,以及对象的交互方式、对象间的关系、集合的工作原理、多态性等内容。另外,还涵盖了一些在其他C#书中很难见到的优秀主题,如对象建模、通过用例格式化需求、UML建模、文件持久化等。

本书涵盖了面向对象开发的方方面面。穿插在全书中的代码示例非常具有解释性，它们将作者想要传达给读者的意图淋漓尽致地呈现出来，成为作者与读者之间的粘合剂。

代码示例、图表、注意和练习使读者的学习体验更轻松，他们无需再陷入永无止境的理论学习中。本书对每个主题的讲解都很透彻，可以当作学习工具和参考书使用。

对于希望进一步提升 C# 技能的任何程序员，我强烈推荐本书！

★★★★☆Great introductory book for OOA/D, December 14, 2008

By N. Venkata "TechTalkToe" (VA, USA)

I have over 6 years of programming experience in procedural languages and was looking for a good introductory book for OOA/D in a OOP language such as C#. This book is a great start.

Pros: Great introduction to concepts and the topics are well structured and presented. Tends to be a bit specific to C# in the 3rd part (code implementation of the design topics).

Cons: I wished they had solutions to the excersies at the end of each chapter.

在过程化语言编程方面，我有超过 6 年的经验，我一直期盼着在面向对象编程语言(如 C#)的 OOA/D 方面有一本能够介绍概念的优秀书籍，这本书正好能够成为我极好的跳板。

优点：书中对概念和主题的介绍在结构上十分清晰。第III部分(设计主题的代码实现)更具体地介绍了 C#。

缺点：我希望本书每章末尾都提供练习题的答案，供读者参考。

★★★★★Excellent, April 22, 2009

By Lcole48(Bensalem, PA USA)

I am a VB.NET developer and needed to get up to speed on C# language and designing application with C# in mind. This book has helped significantly in a very short period of time. I highly recommend it to anyone transitioning from other languages and development environments.

我是一名 VB.NET 开发人员，在我希望转而使用 C#来设计应用程序时，我惊喜地发现了本书，如获至宝，因为该书及时地助我过渡到 C#语言。对于希望从其他语言和开发环境迁移到 C#语言的任何程序员，我都强烈推荐本书！

作者简介



Grant Palmer 在位于加利福尼亚 Moffett Field 的 NASA(美国航空航天局)艾姆斯研究中心的空间技术部门任职超过 23 年。他担任 NASA 的工程师超过 15 年, 并且目前在 ELORET 公司担任科学程序员, 负责开发航天器重新进入地球大气层时的热保护系统的计算机应用程序。

Grant 在加利福尼亚大学的伯克利分校获得了机械工程学士学位, 之后又在斯坦福大学获得航空和航天工程硕士学位。Grant 是 FORTRAN、C、C++ 和 Perl 方面的专家, 但他现在更多地使用 Java 和 C# 等现代化语言进行编程。他已经编写和与其他人合作编写了 7 本计算机编程方面的书籍, 包括 *Physics for Game Programmers* (Apress) 和 *C# Programmer's Reference* (Wrox)。

Grant 与妻子 Lisa 一起居住在华盛顿大学的巴索校区。他有两个儿子 Jackson 和 Zachary, 并且饲养了一些宠物。



Jacquie Barker 是专业的软件工程师和作家, 并且是位于弗吉尼亚州费尔法克斯的乔治梅森大学(GMU)以及位于华盛顿特区的乔治华盛顿大学的编外教师。Jacquie 具有超过 25 年的软件工程和项目管理经验, 她在过去 12 年中关注对象技术, 并且成为对象建模师和 Sun Microsystems 认证的 Java 开发人员。Jacquie 当前在位于弗吉尼亚州荷顿市的 Technology Associates, Inc. 公司工作, 是技术团队的资深成员。

同时, 她也是对象技术咨询和培训机构 ObjectStart LLC 的创始人。

Jacquie 在俄亥俄州克利夫兰市的 Case Western Reserve 大学以最高荣誉获得计算机工程学士学位, 之后又在 UCLA(哥伦比亚大学洛杉矶分校)获得计算机科学硕士学位(研究方向是软件系统工程), 随后继续在 GMU 参加信息技术的研究生课程。

Beginning C# Objects 的第 1 版改编自 Jacquie 最畅销的书籍 *Beginning Java Objects: From Concepts to Code*, 该书最初由 Wrox Press Ltd. 出版, 现在由 Apress 出版。Jacquie 传授对象基础技术的“成功方程式”获得了世界范围内无数读者的赞誉, 而 *Beginning C# Objects: From Concepts to Code* 也被许多大学用作核心 IT 课程的主要教科书。ObjectStart Press 现在已经出版了 Jacquie 的最新书籍 *Taming the Technology Tidal Wave: Practical Career Advice for Technical Professionals*。

请访问 Jacquie 的网站 <http://objectstart.com> 和 <http://techtidalwave.com>, 从中可以获得关于其出版的书籍和提供的服务的更多信息。

在日常生活中, Jacquie 深爱她的丈夫 Steve; 4 只宠物鼠: Kwiddie、Tiffie、Walter 和 Wynxie; 以及作为动物援救志愿者的工作(请访问 <http://petsbringjoy.org>)。在参加计算机相关工作之前, Jacquie 和 Steve 喜欢骑摩托车遍游弗吉尼亚乡村、骑双人自行车以及与家人和朋友一起共度美好时光。

技术编辑简介



Andy Olsen 是在英国工作的自由开发人员和顾问。从 Beta 1 版本面世开始, Andy 就一直在使用 .NET, 并且为 Apress 编写和编辑了多本相关书籍, 涉及范围包括 C#、Visual Basic、ASP.NET 和其他主题。Andy 是狂热的足球和橄榄球爱好者, 并且喜欢跑步和滑雪(尽管水平很糟糕)。他与妻子 Jayne 以及两个孩子 Emily 和 Thomas 一起居住在英国斯旺西市的海边, Emily 和 Thomas 都非常热爱冲浪!

序 言

作为 Java 开发人员和讲师的 Jacquie Barker, 为了表达自己对于彻底学习对象技术是掌握一种面向对象编程语言(OOPL)的第一步的狂热信仰, 精心编写了第一本书籍 *Beginning Java Objects*。自从该书在 2000 年 11 月份出版以来, 我们已经从无数读者处获得了全心全意的认可!

因此, 当 Apress 的出版人 Gray Cornell 和 Apress 编辑部主任 Dominic Shakeshaft 盛情邀请我们编写 *Beginning Java Objects* 的 C#版本时, 我们非常乐意地接受了这个邀请。基本的对象概念都是与语言无关的, 这一点千真万确。在本书的第 I 部分中将介绍对象的相关概念, 在第 II 部分中将介绍对象建模技术。这些对象概念和对象建模技术可以同等地应用于任何 OOPL。

但是, 本书有两个目标: 我们不仅希望介绍对象概念和对象建模技术, 而且希望通过展示如何专门将这些概念转换为 C#语法, 帮助读者正确地掌握 C#编程语言。基于这两个目标, 我们编写了 *Beginning C# Objects* 一书!

Beginning C# Objects 的第 1 版发布于 2004 年春天。在此之后, C#和.NET 平台已经发生了巨大的变化。Microsoft 已经发布了 .NET Framework 的多个新版本, 并且在 C#编程语言中增加了许多令人兴奋的新功能。本书的第 2 版全面介绍 .NET 和 C#中适合于编程新手的最新且最优秀的功能, 帮助读者成为专家级的 C#程序员。

Grant Palmer 和 Jacquie Barker

前言

首先, *Beginning C# 2008 Objects: From Concepts to Code* 是一本关于软件对象的书籍, 全面介绍了对象的概念, 对象如此神奇而直观的原因以及程序员如何构造软件应用程序以正确地使用对象。

本书是一本关于 C# 的书籍, 但不是一本“C# 大全”, 而是对该语言的和缓而全面的介绍。本书特别强调如何将对象模型转换为具备完整功能的 C# 应用程序, 其他书籍很少提供这方面的介绍。

本书目标

编写本书的目标(希望也是您购买本书的目的)如下:

- 帮助读者熟悉基本的面向对象(object-oriented, OO)术语和概念。
- 提供对象建模的亲手实践的体验; 也就是说, 开发可用于随后构建 OO 软件系统的基础的“蓝图”。
- 展示如何将对象模型转换为可用的软件应用程序的基础——特别是对于 C# 应用程序, 虽然本书介绍的对象建模技术可以同等地应用于任何面向对象编程语言(object-oriented Programming language, OOPL)。

如果您已经使用过 C# 语言(但是不熟悉有关对象的基础知识), 本书将提供关于该语言的 OO 根源的关键知识。另一方面, 如果您是 C# 初学者, 本书将帮助您正确地起步。从这两方面来看, 对于希望精通 OOPL(如 C#)的任何人, 本书都将是必须阅读的书籍。

同样重要的是, 本书并不打算实现如下目标:

- 帮助读者快速精通对象建模: 类似于所有高级技能, 完全掌握对象建模技术有两个前提条件: 良好的理论基础和大量实践。本书提供了基础知识, 以及帮助您应用和实践所学新知识的项目和练习的建议。但是, 真正掌握对象建模的唯一方法是不断参与 OO 建模和开发项目。本书中提供的各种技能(希望能够给您带来信心), 帮助您在专业机构中开始应用对象技术, 这样您才能真正学习对象技术。特别是, 如果有一位 OO 经验丰富的导师在您的第一次符合“行业标准”的项目中不断指导您, 您将受益匪浅。
- 帮助读者成为特定 OO 方法论方面的专家: OO 软件开发领域有数十种不同的正式方法; 新的方法不断出现, 而且并无优劣之分。例如, UML(Unified Modeling Language, 统一建模语言)符号体系是一种最新的方法, 而 OMT(Object Modeling

Technique, 对象建模技术)符号体系是一种最古老的方法, 然而两者极为相似, 因为 UML 从根本上极大地扩展了 OMT。确保理解对象建模的一般过程以及 UML 的细节之后, 您就具备了足够的知识, 可以自己检阅、评估和选择特定的方法论(或者创建自己的方法论——甚至将来某一天编写关于自己创建的方法论的书籍)。

- 介绍读者所需的全部 C#知识: C#是内容相当丰富的语言, 由数十个核心类、可以通过 .NET 类库使用的数百个其他类以及由这些类执行的数以千计的操作组成。如果 C#提供了执行特定操作的多种备选方法, 本书将只介绍一到两种最适合于解决当前问题的方法, 帮助读者理解解决该问题的方式。尽管如此, 本书仍然提供了构建完整的应用程序所需的足够 C#语言知识。

掌握从本书中获得的知识之后, 您就能够理解关于 C#的更为彻底的阐述, 例如, 当前市面上的其他 C#参考书籍提供的知识或者深入介绍 UML 的参考书籍中的知识。

成功的 OO 程序员必须理解对象的原因

我们经常与一些软件开发人员会面——在工作场所、在客户的办公室、在专业会议上以及在大学校园内——这些开发人员都尝试掌握 C#这样的 OOPL, 他们参加 C#培训, 阅读 C#相关书籍, 或者安装并使用 C#集成开发环境(integrated development environment, IDE), 如 Visual Studio .NET。然而, 这是舍本逐末的方法: 他们缺乏对象概念的基础知识, 更严重的是, 他们不知道如何从头开始构造软件应用程序以充分利用对象。

假设您知道如何构造住宅的基础知识, 并且被要求构建房屋。实际上, 您还是一位广受欢迎的世界知名建筑家! 客户告诉您, 建筑材料已经运送给您。在开始构造房屋的当天, 一辆卡车开到了施工现场, 卸下一大堆蓝色、星型的奇怪建材, 每块建材的中间有一个空洞。因此, 您只有完全停工! 您曾经使用木材、砖材和石材等建筑材料建构过无数房屋, 并且知道如何使用熟悉的材料施工; 但是, 您并不知道如何使用蓝星材料堆叠房屋。

经过一番冥思苦想, 您拿出一把锤子和一些钉子, 尝试按照以前处理木材的方式将这些蓝星材料钉在一起, 但是材料无法完全合缝。然后, 您又尝试使用砌砖材的灰泥填充蓝星材料之间的缝隙, 但是灰泥似乎无法将这些蓝星材料粘在一起。由于预算紧张并且工期有限制(并且由于您自称是建筑专家, 但是却不知道如何使用现代化的建材), 因此您只好勉强施工。最终, 您拼凑出一个至少在外表上看起来类似于房屋的建筑。

您的客户前来检查工作, 他完全感到失望。客户选择蓝星材料作为建材的一个原因是它们相当节省能源, 但是因为您使用钉子和灰泥装配蓝星建材, 它们失去了大部分隔离房屋的固有能量。为了补偿热量的损失, 客户要求您将房屋中的窗户全部换为绝热玻璃窗, 这使您手足无措! 更换窗户花费的时间远多于新建房屋的时间, 更不要说替换在更换窗户时损坏的蓝星材料所需的成本。当您告诉客户目前的情况时, 他立刻暴跳如雷! 客户选择蓝星材料作为建材的另一个原因是, 这种建材具有很强的适应性, 易于迎合设计的更改, 但是由于您采用无效的方式装配蓝星材料, 因此必须逐个拆开它们, 并更换

其中的大部分。

遗憾的是，这也是许多程序员构建 OO 应用程序的方式，因为他们并没有很好地理解如何从对象的角度完成项目。更糟糕的是，绝大多数即将成为 OO 程序员的人员对于理解对象并将其用于 OO 语言编程的必要性一无所知。因此，他们使用 C#这样的语言编程拼凑出远非理想的结果：在必须执行“中期修正”（例如，在部署应用程序之后需要引入新功能）时缺乏灵活性的程序。

本书读者对象

本书为希望最有效地使用 C#这样的 OOPL 的任何人编写！本书的目标读者如下：

- 还没有开始使用 C#，但是希望正确起步的人群。
- 曾经购买过 C#书籍并且满怀希望地读完该书籍，但是只理解该语言的基础知识，而不知道如何构造应用程序以充分利用该语言的 OO 功能。
- 已经购买 C# IDE 软件工具，但是实际上只了解如何拖放图形用户界面(graphical user interface, GUI)组件以及在按钮和菜单的后台添加少量逻辑，而没有真正地了解围绕对象正确地构造应用程序的核心。
- 已经构建过 C#应用程序，但是对在应用程序生命周期的后期出现新需求时维护或修改该应用程序的困难程度感到失望。
- 前面已经学习过对象建模的相关知识，但是不了解如何将对象模型转换为真实的代码(C#或其他语言)。

底线是，希望掌握 C#这样的 OO 语言的人必须首先成为对象方面的专家。

为了从本书获得最大的价值，您应该具备一些编程经验，无论何种语言均可。您应理解如下的简单编程概念：

- 简单数据类型(整型、浮点型等)
- 变量和它们的作用域(包括全局数据的概念)
- 控制流程(if-then-else 语句、for/do/while 循环等)
- 数组：数组的概念和使用方式
- 函数/子例程/方法的概念：如何传入数据和返回结果

您不需要预先了解 C#(在本书的第 I 部分将简要介绍该语言，而在第 III 部分将深入研究该语言)。并且，您也不需要了解对象——至少在软件层面上是如此！在第 2 章中将介绍对象，人类天生就是从对象的角度来看世界的。

如果您已经开发过具备完整功能的 C#应用程序，但仍然不清楚应用程序构建中的对象方面，则阅读本书也为时未晚。大学中通常会提供关于对象建模和 C#编程的单独课程。学生最好按照顺序选修这两门课程，但是通常会在选修对象建模之前就已经学习了 C#。这些学生会在本书中看到一些熟悉的“地标(landmark)”(在 C#代码示例中)，但是即使如此，他们仍然会在本书中学到在使用 C#(或其他 OOPL)编程时执行各种操作的原因的基本原理，从而极大地增长见识。

您不仅需要了解语言的机制，更需要了解面向对象的“原因”，这样才能最终成为更

优秀的 C#程序员。如果您已经具备 C#编程经验,则可以快速浏览简单介绍该语言的章节,即第 I 部分中的第 1 章和第 III 部分中的第 13 章。

本书根源于作者的课程教学,因此非常适合于用作一学期课程或分为对象建模和 C#编程两门课程的教科书。

如果您感兴趣于对象建模,但是不希望学习 C#编程

本书仍然绝对有价值!即使您并不打算从事编程行业(许多对象建模学生都是如此),但是阅读少量使用 C#这样的 OOP 编写的代码示例确实有助于您牢固掌握对象概念。因此,鼓励您阅读本书的第 III 部分——至少是第 14 章——即使您永远不会动手编写 C#程序。

本书的结构安排

本书围绕以下 3 个主要主题展开介绍:

第 I 部分: 对象 ABC

在深入研究如何执行对象建模以及 C#的 OO 编程详情之前,我们有必要说同一种关于对象的语言。本书的第 I 部分包括第 1~7 章,从定义位于所有软件开发方式(OO 或其他方式)之下的基础概念开始逐步介绍。但是,第 I 部分很快就开始讨论高级对象概念,从而在学习完这一部分时,您将会充分理解对象。

第 II 部分: 对象建模 101

在第 II 部分(第 8~12 章)中,主要介绍在开发应用程序的对象模型时,执行各种操作的方式以及(更重要的是)原因的原理——这是所有对象建模技术共有的原理。了解 UML 符号非常重要,因为它是行业标准,并且很可能是大多数同事/客户将会使用的符号。因此,本部分将介绍 UML 知识,并且将 UML 用于所有具体的建模示例。使用这些章节中介绍的建模技术,为学生选课系统(Student Registration System, SRS)开发对象模型“蓝图”,本前言的末尾给出了有关该系统的需求说明。

第 III 部分: 将对象“蓝图”转换为 C#代码

在本书的第 III 部分(第 13~17 章)中,将展示如何将第 II 部分中开发的 SRS 对象模型交付为具备完整功能的 C#应用程序,包括图形化用户界面以及在不同用户登录该系统时持久保存数据的方法。这一部分中给出的所有代码示例都可以从 Apress 网站(www.apress.com)和本书信息网站(www.tupwk.com.cn/downpage)下载,并且强烈鼓励您下载这些代码并实验它们。实际上,在每章末尾都提供了可供实验的练习。本部分使用通常表达软件系统需求的描述风格编写 SRS 的需求说明。您可能现在就有信心构建应用程序来解决这种问题,但是在学习完本书之后,您将会更加相信自己能够将其构建为面向

对象的应用程序。

为了完善本书，本书包括了最后一章“下一步”，其中给出了一些建议，帮助您在学习完本书后了解如何继续自己的 OO 发现之旅。本书给出了推荐书籍列表，您可以根据自己下一步的发展领域，借助这些书籍更上一层楼。

作为本书基础的 C#版本

与任何编程语言一样，Microsoft 不断推出新的 C#版本。在出版本书的第 2 版时(2008 年秋天)，.NET 的最新版本是 3.5，因此本书使用了 C#语言的一些“最新和最优秀”的功能，例如，自动实现的属性和对象初始化器。然而，您在阅读本书后获得的正确 OO 编程的技术和概念并不与任何 .NET 版本密切相关，而是同样可以应用于未来的 C#版本。

开始学习本书之前的最后思考

对于有经验的程序员来说，本书中的许多内容——特别是第 I 部分开始处的内容——看起来可能过于简单。这是因为许多对象技术都建立于基础软件工程原理之上，多年以来已经在许多项目中实践过这些原理，在此只是稍微重新包装而已！确实有一些技巧使 OOPL 具备非常强大的功能，而使用非 OO 语言则不可能实现这些功能，例如，分别在第 5 章和第 7 章中介绍的继承和多态(可以在非 OOPL 中手动模仿这些技术，如同程序员可以从头开始编写自己的数据库管理系统(database management system, DBMS)，而不是使用商业化的产品，如 Oracle、Sybase 和 MY SQL Server，但是谁也不愿意这样做)。

对于有经验的程序员，在精通对象的道路上面临的最大挑战是，重新调整他们解决问题的方式：

- 已经使用非 OO 方法开发过应用程序的软件工程师/程序员通常必须“忘却”在软件分析和设计的传统方法中使用的特定手段。
- 看似矛盾的情况是，对于刚刚成为程序员(或者成为 OO 建模师)的人，他们在学习软件开发的 OO 实现时反而更容易接受和用于实践。

幸运的是，在开发软件时思考对象的方式基本上与人们通常思考世界的方式相同。因此，学习“思考”对象以及使用 C#编写对象就像 1、2、3 一样容易！

读者反馈

我们努力编写本书，力求使本书为读者提供更多的有用信息，因此希望知道您的想法，并且始终急切地希望了解您需要知道的信息。

我们衷心感谢您对本书提供的反馈，并且在将来的编辑工作中牢记您所提出的批评和建议。如果您希望给出反馈意见，可以在 info@apress.com 或 www.apress.com 上直接发布反馈意见，或者通过 grantpalmer@gmail.com、jacquie@objectstart.com 或

<http://objectstart.com> 与作者联系。

SRS(学生选课系统)需求说明书

STUDENT REGISTRATION SYSTEM (SRS) CASE STUDY: STUDENT REGISTRATION SYSTEM REQUIREMENTS SPECIFICATION

We have been asked to develop an automated Student Registration System (SRS) that will enable students to register online for courses each semester, as well as track a student's progress toward completion of his or her degree.

When a student first enrolls at the university, he or she uses the SRS to set forth a plan of study about which courses he or she plans to take to satisfy a particular degree program, and chooses a faculty advisor. The SRS will verify whether the proposed plan of study satisfies the requirements of the degree the student is seeking. Once a plan of study is established, students are can view the schedule of classes online during the registration period preceding each semester and then choose whichever classes they wish to attend, indicating the preferred section (day of the week and time of day) if the class is offered by more than one professor. The SRS will verify whether the student has satisfied the necessary prerequisites for each requested course by referring to the student's online transcript of courses completed and grades received (the student can review his or her transcript online at any time).

Assuming that (a) the prerequisites for the requested course(s) are satisfied, (b) the course(s) meets one of the student's plan-of-study requirements, and (c) there is room available in each of the class(es), the student is enrolled in the class(es).

If (a) and (b) are satisfied, but (c) is not, the student is placed on a first-come, first-served waiting list. If a class/section that he or she was previously waitlisted for becomes available (either because some other student has dropped the class or because the seating capacity for the class has been increased), the student is automatically enrolled in the waitlisted class, and an email message to that effect is sent to the student. It is his or her responsibility to drop the class if it is no longer desired; otherwise, he or she will be billed for the course.

Students can drop a class up to the end of the first week of the semester in which the class is being taught.

目 录

第 I 部分 对象 ABC

第 1 章 初步体验 C#	3
1.1 C#入门指南	4
1.2 使用 C#的原因	4
1.2.1 实践出真知	4
1.2.2 C#是集成应用开发架构的一部分	4
1.2.3 C#是彻底的面向对象语言	5
1.2.4 C#是免费的语言	5
1.3 C#语言基础	6
1.4 详细分析一个简单的 C#程序	6
1.4.1 “using System;” 语句	7
1.4.2 注释	7
1.4.3 类声明/“包装器”	8
1.4.4 Main 方法	8
1.5 预定义类型	9
1.6 变量	10
1.6.1 变量的命名约定	11
1.6.2 变量初始化和赋值	11
1.7 字符串	12
1.8 区分大小写	13
1.9 C#表达式	13
1.9.1 赋值语句	13
1.9.2 算术运算符	14
1.9.3 求值表达式和运算符优先级	16
1.9.4 逻辑运算符	16
1.10 隐式类型转换和显式类型转换	18
1.11 循环和其他流程控制结构	19

1.11.1 if 语句	19
1.11.2 switch 语句	22
1.11.3 for 语句	25
1.11.4 while 语句	28
1.11.5 do 语句	29
1.12 跳出语句	29
1.13 代码块和变量作用域	31
1.14 输出到屏幕	32
1.14.1 Write 和 WriteLine 的对比	34
1.14.2 转义序列	35
1.15 C#样式的要素	36
1.15.1 适当地使用缩进	36
1.15.2 明智地使用注释	39
1.15.3 大括号的放置	40
1.15.4 自说明的变量名	42
1.16 本章小结	42
1.17 练习	43
第 2 章 抽象和建模	45
2.1 通过抽象进行简化	45
2.2 通过抽象进行概括	46
2.2.1 将抽象模型组织为分类层次结构	47
2.2.2 作为软件开发基础的抽象	50
2.3 抽象的重用	51
2.4 固有的挑战	51
2.5 本章小结	53
2.6 练习	54

第3章 对象和类55	4.4 信息隐藏/可访问性90
3.1 对象的概念.....55	4.4.1 可访问性的类型.....91
3.2 状态/字段/数据.....56	4.4.2 在类方法内部访问类的成员.....95
3.3 行为/操作/方法.....57	4.4.3 再论 Camel 命名法和 Pascal 命名法.....96
3.4 类.....58	4.4.4 再论方法头.....97
3.4.1 关于命名约定的说明.....59	4.4.5 从客户代码访问私有成员.....97
3.4.2 实例化.....59	4.4.6 字段值的持久性.....104
3.5 用户定义类型和引用变量.....61	4.4.7 公有/私有规则的例外.....104
3.6 实例化对象: 深入研究.....62	4.5 封装的强大功能107
3.7 作为字段的对象.....65	4.5.1 防止对封装数据未经授权的 访问.....107
3.8 关联.....66	4.5.2 帮助确保数据完整性.....108
3.9 面向对象编程语言的 3个特性.....68	4.5.3 当私有成员更改时限制 连锁反应.....109
3.10 本章小结.....69	4.5.4 在类的自身方法中使用 存取器.....112
3.11 练习.....69	4.6 构造函数114
第4章 对象交互71	4.6.1 默认构造函数.....114
4.1 事件驱动对象协作.....71	4.6.2 编写自己的构造函数.....115
4.2 声明方法.....72	4.6.3 向构造函数传递参数.....116
4.2.1 方法头.....73	4.7 本章小结118
4.2.2 向方法传递参数.....73	4.8 练习118
4.2.3 方法返回类型.....74	第5章 对象间的关系121
4.2.4 方法体.....75	5.1 关联和链接.....121
4.2.5 方法实现业务规则.....76	5.1.1 多样性.....123
4.2.6 return 语句.....76	5.1.2 聚集.....125
4.2.7 命名建议.....78	5.1.3 继承.....126
4.3 方法调用和点记号法.....79	5.1.4 继承的优点.....133
4.3.1 实参与形参.....79	5.1.5 继承的缺点.....133
4.3.2 作为方法调用上下文的对象.....80	5.1.6 类的层次结构.....134
4.3.3 再论 C#表达式.....82	5.1.7 继承是否真是一种关系.....135
4.3.4 获取方法调用的返回值.....82	5.1.8 避免“连锁反应”.....136
4.3.5 方法签名.....83	5.1.9 派生类的规则: 可为.....136
4.3.6 对象通过方法交互.....84	5.1.10 派生类的规则: 不可为.....140
4.3.7 通过点记号法访问字段.....85	5.1.11 重载.....141
4.3.8 委托.....86	5.2 略谈多重继承144
4.3.9 访问对象.....87	
4.3.10 作为客户和服务提供者 的对象.....89	

5.3 回顾面向对象编程语言的 3 个显著特点	147	7.3 抽象类	185
5.4 本章小结	147	7.3.1 抽象类和实例化	188
5.5 练习	148	7.3.2 重写抽象方法	189
第 6 章 对象集合	151	7.3.3 打破抽象的咒语	190
6.1 集合的定义	151	7.4 接口	192
6.1.1 在初次使用集合之前 必须对其进行实例化	152	7.4.1 实现接口	193
6.1.2 通过类定义集合	154	7.4.2 “是一个”关系的 另一种形式	195
6.1.3 封装 OO 集合	154	7.4.3 抽象类与接口的对比	195
6.2 作为简单集合的数组	155	7.4.4 实现多个接口	197
6.2.1 声明和实例化数组	155	7.4.5 接口和实例化	200
6.2.2 访问单个数组元素	155	7.4.6 接口的重要性	200
6.2.3 初始化数组内容	156	7.5 静态成员	203
6.2.4 操作对象数组	157	7.5.1 静态字段	203
6.2.5 其他数组考虑事项	159	7.5.2 静态属性	207
6.2.6 多维数组	160	7.5.3 静态方法	208
6.3 更复杂的集合类型	162	7.5.4 静态方法和属性的限制	210
6.3.1 有序列表	163	7.5.5 C#的特有术语	212
6.3.2 组	164	7.6 实用程序类	212
6.3.3 字典	165	7.6.1 用户定义的实用程序类	213
6.3.4 泛型集合	166	7.6.2 常量	215
6.3.5 多个集合同时引用 同一个对象	167	7.6.3 只读字段	217
6.4 作为方法返回类型的集合	168	7.7 本章小结	217
6.5 超类型的集合	170	7.8 练习	218
6.6 再论组合类	171	第 II 部分 对象建模 101	
6.6.1 courseLoad	171	第 8 章 对象建模过程简述	223
6.6.2 transcript, 方法一	172	8.1 对象建模的目标	223
6.6.3 transcript, 方法二	173	8.2 对象建模过程概述	226
6.6.4 完整的 Student 数据结构	174	8.2.1 关于对象建模软件工具 的思考	227
6.7 本章小结	174	8.2.2 提醒	229
6.8 练习	175	8.3 本章小结	229
第 7 章 多态和最后一些对象概念	177	8.4 练习	229
7.1 多态的定义	177	第 9 章 通过用例规范化需求	231
7.2 OOPL 的 3 种与众不同 的功能	184	9.1 用例的概念	231
		9.1.1 功能需求和技术需求	232