

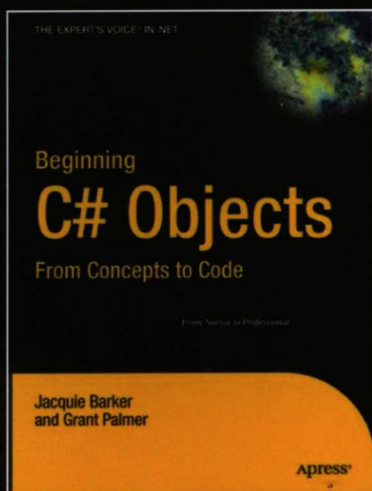
Beginning C# Objects

THE EXPERT'S VOICE® IN .NET

中文
版

From Concepts to Code

概 念 到 代 码



From Novice to Professional

[美] Jacquie Barker
Grant Palmer
韩磊 戴飞

著
译



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Beginning C# Objects

中文版

——概念到代码

Beginning C# Objects: From Concepts to Code

[美] Jacquie Barker 著
Grant Palmer
韩磊 戴飞 译

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书关注的是软件对象：什么是软件对象？为什么它们如此神奇而直白，以及如何正确使用对象去构建一个软件应用。这也是一本关于 C# 的书，但它并非一本“C# 大全”，而是一份对该语言的温文但全面的介绍，还特别强调了如何把对象模型转换到功能完备的 C# 应用程序。本书分三部分，结合 C# 深入浅出地阐述面向对象理论的基本内容，即从对象基本概念到对象建模，到 C# 代码的整个学习过程。为了完善本书，我们还写了标题为“下一步”的最后章节，它给出一些建议，让你在读完本书后，懂得如何继续自己的面向对象发现之旅。

Original English language edition published by Apress L. P., 2560 Ninth Street, Suite 219, Berkeley, CA 94710 USA. Copyright © 2004 by Apress L. P. Simplified Chinese-language edition copyright © 2006 by Publishing House of Electronics Industry. All rights reserved.

本书简体中文专有翻译版权由 Apress L. P. 公司授予电子工业出版社，未经许可，不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号： 图字：01-2004-2486

图书在版编目（CIP）数据

Beginning C# Objects 中文版：概念到代码 / （美）芭可（Barker, J.），（美）帕尔默（Palmer, G.）著；韩磊，戴飞译. —北京：电子工业出版社，2006.6

书名原文：Beginning C# Objects: From Concepts to Code

ISBN 7-121-02493-4

I. B... II. ①芭...②帕...③韩...④戴... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字（2006）第 035905 号

责任编辑：周筠 梁晶

印 刷：北京智力达印刷有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销：各地新华书店

开 本：787×1092 1/16 印张：42.5 字数：800 千字

印 次：2006 年 6 月第 1 次印刷

定 价：68.00 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。质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

技
术
凝
聚
实
力

Broadview[®]
WWW.BROADVIEW.COM.CN

专
业
创
新
出
版

博文视点

- 与向上的心合作，共同成长 -

译者序

入门不简单

由于工作性质的关系，我常常需要为公司面试程序员。通常，我会首先要求应聘者做两件事：第一，口头说明虚方法、抽象方法、接口之间的异同和使用场景；第二，脱离 IDE，手写一个简单的 WinForm 程序（例如包括布局整齐的两个文本框和三个按钮、点击按钮弹出消息框显示文本框内容），然后在命令行编译运行。第一题是考面向对象中的继承和多态概念，第二题则是考 C# GUI 类和关于委托等 C# 独有特性的基础知识，以及编译、调试技巧。很遗憾，有一大半的应聘者会在这两个简单考题面前败下阵来。尤其是第二个考题，甚至难倒了一些写代码有年头的程序员。为什么？因为这些程序员不具备基本的面向对象知识，更加不理解 C# 的面向对象实现手法。

这不能完全归咎于中国计算机教育。实际上，在美国或其他发达国家，真正懂得面向对象开发的程序员也并非随处可见。这也是 *Beginning C# Objects*（及其 Java 原版 *Beginning Java Objects*）一书成为畅销书、并在 Amazon 上获得读者极高评价的原因。作者写道：“我们常常与一些软件开发者会面——在工作场所、在客户办公室、在专业会议上，或在大学校园里——这些开发者都尝试去掌握一门类似 C# 的 OO 编程语言，他们参加 C# 培训、阅读关于 C# 的书，或是安装和使用像 Visual Studio .NET 这样的 C# 集成开发环境（IDE）。然而，这是舍本逐末的做法：他们缺乏对什么是对象的基础认识，更为严重的是，缺乏利用对象从头开始构建软件应用程序的知识。”

翻开市面上任何一本你能找到的 C# 入门书，看看第一章讲什么，第一个范例是怎么实现的。闭上眼睛你都能告诉我，第一章多半是讲怎么安装 VS.NET，第一个例子多半是拖一个按钮控件到窗体，双击后输入一段调用消息框的代码。我们的技术作者们，就是这样把读者引入歧途的。IDE（集成开发环境）能极大地提升生产力，但开发应用程序所需的高度专业的知识和技能，却非 IDE 所能代替。的确，任何一个菜鸟也许都能利用可

可视化组件拼凑出“看起来还不错”的应用程序,但这样的程序却将带来高昂的扩展和维护成本。

面向对象编程,绝非一些可视化组件那么简单,它涉及人类思维(抽象)模式、建模符号体系、面向对象方法学等诸多方面,对开发者有较高的要求。功夫过关的开发者,不止是技术高手,同时也一定会是有深度的思想者。把需求从自然语言翻译为对象模型,再把对象模型翻译为特定语言代码,殊非易事。最基础的,到底什么是对象?如果你曾经好好思考过这个问题,就会得到很多启发。例如,现实世界中的一张纸,如何抽象为计算世界中的对象?这个对象将具有哪些特性(属性和行为),例如尺寸、颜色、质地、折叠、裁减、卷筒……?综合来看,面向对象的要素是什么?把事物抽象为对象的过程,是做思维体操的过程,也是极富挑战性和乐趣的过程。如果你还不了解对象和面向对象的概念,或想与作者一起就该话题做更深入的思考,那本书第一部分就是最好的入门手册。

我认识一些优秀的程序员朋友,他们在大学时念的专业是建筑。这些朋友对应用程序架构和/或开发流程,有近乎严苛的要求,因为他们深刻地理解,蓝图、材料、工序……对于建造房屋是多么的重要。对象模型的静态方面(域类、数据结构等)在应用程序中开发相当于建材单元,而动态方面(行为、方法等)则是关于建材之间如何组合的指导书,它们合起来,构成一个应用程序的“蓝图”。越是复杂、大型的应用程序,对蓝图的要求就越高;即便是简单的应用,有蓝图也比较有利于维护、升级和扩展。给你一份需求说明书,你将如何分析它、并且组织出正确的对象模型(蓝图)?如果你对此信心不足,建议好好阅读本书第二部分。

第三部分涉及的范例,在有经验的开发者眼中看来,似乎过于简单。其实不然。整个开发过程当中,没有使用 IDE 拖放过任何一个控件,或在 IDE 中编译调试。这样做的目的有二:第一,让读者可以掌握.NET Framework 和 C#本身的特性,而不会被 IDE 的花哨界面所迷惑、急于求成;第二,帮助读者学会用正确的手法和模式(如公认的 MVC 模式)开发程序。例如委托(delegate),这是.NET Framework 中一种特别的语言元素,也不易理解。如果你只懂得往窗体放一个按钮,双击该按钮,输入一行代码,那么你永远不会明白,这个过程体现了利用委托实现事件处理的巧妙手法。但是,如果让你脱离 IDE 编写事件处理方法,你就很快能明白这个道理,而且也知道怎么用于实践。IDE 能提升生产力,不过它却不能凭空创造生产力,本源的生产力还是来自开发者的知识与技能。

面向对象编程已经流行多年，然而还是有无数的入门者在入门阶段就走错路子，抱着错误的观念、用错误的方式开发着意大利面般一团乱麻的程序。入门不简单，对于初学者如此，对于有经验的开发者，更该回头检讨自己在面向对象编程领域的经验是否根本就是错误或细枝末节的经验。本书作者开发和培训经验丰富（一位是 NASA 开发工程师，一位是大学教师 and 对象技术专家），理解面向对象程序员可能进入的误区和面对的挑战，他们知识与经验的总结，形成了这么一本用心良苦、循循善诱、深入浅出的 C# 面向对象开发指南。每一位 C# 开发者，无论有经验还是没经验，都该读读这本书。入门不简单，你真的入对门了吗？

韩 磊

2006 年 5 月于北京

关键术语译法中英对照表

下表列出了本书用到的主要术语的英文原文以及对应的中文译法，供读者参考和辨析。

英文术语	中文译法
abstract class	抽象类
abstract method	抽象方法
accessibility	可访问性
accessor method	访问器方法
aggregation	聚合
ancestor class	祖先类
anonymous object	匿名对象
argument	参数值
array	数组
association class	关联类
attribute	(不翻译)
base class	基类
binary association	二元关联
candidate class	候选类
collection	群集
common supertype	公共超类
concrete class	具体类
constant	常量
container	容器
data dictionary	数据字典
data integrity	数据完整性
data member	数据成员
data type	数据类型
debug	调试
delegate	委托
derived class	派生类
domain class	域类
domain objects	域对象
dot notation	点符号
dynamic model	动态模型
encapsulation	封装

英文术语	中文译法
enumeration	枚举
escape sequence	转义序列
event delegation	事件委托
event driven	事件驱动
event handling	事件处理
exception	异常
explicit casting	显式转换
field	字段
flow control	流程控制
form (GUI)	窗体
garbage collection	垃圾收集
get accessor	取值访问器
get method	取值方法
get/set method	取值/赋值方法
handle	句柄(名词)
implicit casting	隐式转换
inheritance	继承
instance constructor	实体构造器
instantiation	实体化
interface	接口
jagged array	锯齿状数组
key-value pair	名-值对
member	成员
message	消息
metadata	元数据
method	方法
method header	方法头
method hiding	方法隐藏
method signature	方法签名
modeling	建模

英文术语	中文译法
modifier	修饰符
multiplicity	多重性
namespace	命名空间
noun phrase analysis	名词短语分析
ordered list	有序列表
overload	多载
override	覆载
parameter	参数
persistence	持久化
polymorphism	多态
predefined types	预定义类型
private	私有的
property	(不翻译)
protected	受保护的
public	公布的、公共的
reflexive	反身
return type	返回类型
sealed	封闭的
sequence diagram	时序图
set (collection)	集合
set accessor	赋值访问器
set method	赋值方法
sibling class	兄弟类
static	静态的
superclass	超类
use case	用例
variable	变量

In loving memory of my wonderful parents, Bill and Dorothy Jost.

谨以此纪念我的佳父良母，*Bill Jost* 和 *Dorothy Jost*。

——*Jacquie Barker*

I would like to dedicate this book to my dog Bailey, who is a good, old dog.

我要将本书献给我的老好狗 *Bailey*。

——*Grant Palmer*

About the Authors

关于作者



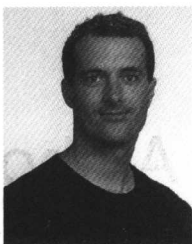
Jacquie Barker 是一位专业软件工程师、作者和华盛顿特区乔治·华盛顿大学编外教师。Jacquie 拥有超过 25 年的软件开发和相关管理经验，在过去 12 年中集中关注对象技术，并成为对象建模师和 Sun Microsystems 认证 Java 程序员。她目前在位于弗吉尼亚州荷顿县的 Technology Associates, Inc. 公司工作，任技术班子资深成员。她同时也是对象技术咨询和培训机构 ObjectStart LLC 的创始人。

Jacquie 在俄亥俄州的 Case Western Reserve 大学以最高等级荣誉获理学学士学位。稍后，又获 UCLA（哥伦比亚大学洛杉矶分校）的计算机科学硕士学位，研究方向是软件系统工程。之后，她继续参加位于弗吉尼亚州 Fairfax 的 George Mason 大学的研究课程。

Beginning C# Objects 改编自 Jacquie 的畅销书，Beginning Java Objects: From Concepts to Code。该书初版由 Wrox Press, Ltd. 出版，现由 Apress 出版。Jacquie 传授对象基础技术的“成功方程式”持续得到世界各地读者的赞誉，而 Beginning Java Objects 一书也被多个大学采用为核心 IT 课程的主要教科书。

请访问 Jacquie 的网站，<http://objectstart.com>，获得她的另外一些著作和服务的更多信息。

在生活中，Jacquie 热爱她的丈夫 Steve，也热爱宠物鼠、巧克力和 Java 编程。在参加研究课程之前，她热衷于与 Steve 一起骑摩托车遍游弗吉尼亚乡村，骑双人单车、海滩度假，与家人和朋友一起共度好时光。



Grant Palmer 在位于加利福尼亚 Moffett Field 的 NASA (美国航空航天局) 艾姆斯研究中心空间技术部供职超过 20 年。Grant 任 NASA 的工程师达 15 年之久, 目前在 ELORET 公司 (NASA 的一个关联公司) 任科学程序员, 开发用于帮助设计航天器重入地球大气层时的热保护系统的计算机程序。

Grant 在加利福尼亚大学伯克利分校获机械工程学士学位, 稍后, 他又在斯坦福大学获航空和航天工程硕士学位。Grant 是 FORTRAN、C、C++ 和 Perl 语言的专家, 但最近多用 Java 和 C# 等更现代的语言编写程序。他撰写或与人合写了七本关于计算机编程的著作, 其中包括 *C# Programmer's Reference* 一书。

Grant 与妻子 Lisa 一起住在亚利桑那州的凤凰城。他有两个儿子, Jackson 和 Zachary, 还有一条名叫 Bailey 的老狗。

Acknowledgments

致 谢

我愿诚挚、衷心地感谢那些帮助 Grant 和我完成本书的人：

- 谢谢 James Huddleston，他是一位真正的才子和非凡的评论家，作为我们的第一技术审校，他帮助我们列出了 C# 的“妙处”，居功甚伟。
- 谢谢 Dominic Shakeshaft，我们的编辑，他的努力使本书脉络清晰。
- 谢谢 Gary Cornell，Apress 出版人，他提出了把 *Beginning Java Objects* 一书“移植”到 C# 版本的建议。
- 谢谢所有 Apress 的同事——特别是 Kylie Johnston、Ami Knox、Laura Cheu 和 Glenn Munlawin——谢谢他们在编校/印刷/营销过程中的大力支持。
- 谢谢我的合作者 Grant Palmer，他在“最黑暗的日子”里也保持了最好的幽默感！
- 谢谢我的丈夫 Steve，当我又暂时陷入“写书大计”中时，你再次显示出耐心。我对你的爱数以吨计！！！！

—Jacquie Barker

谢谢我的合作者 Jacquie Barker，和她共事令人愉快，她也教给我很多写作和面向对象原理的知识。

谢谢技术审校者——Jim Huddleston 和 Dominic Shakeshaft——他们给本书提出了极好的评论和建议。Jim 是一位真正的计算机大师，他忘掉的编程知识可能比我知道的多得多。

谢谢 Apress 的编校/印刷班子——Kylie Johnson、Ami Knox、Laura Cheu 和 Glenn Munlawin——他们不知疲倦的工作，使本书能保质保量完成。

最后，谢谢我美丽的妻子 Lisa，以及我们的好儿子 Jackson 和 Zachary，你们让生命值得过活。

—Grant Palmer

Preface

序

作为 Java 开发者和讲师，我撰写了我的第一本书《Beginning Java Objects》表达我关于彻底学习对象技术是掌握一门面向对象编程语言最基础的第一步的狂热信仰。自该书于 2000 年 11 月出版以来，我从无数读者处得到了全心全意的认同。

我为 B.J.O. 所获得的良好反馈而倍感欣悦，所以，当 Gray Cornell, Apress 的出版人，以及 Dominic Shakeshaft, Apress 编辑部主任，邀请我撰写本书的 C# 版本时，我也非常乐意。基本的对象概念是“语言中立”的，这千真万确。你将在本书第一部分学到的对象概念，以及在第二部分学到的对象建模技术，可应用于 C#、Java、Visual Basic .NET、C++、Ada 或 Smalltalk，或者任何尚未发明的面向对象语言。

不过，这本书有双重目的：我们不只是想教给你关于对象和对象建模的概念，还想通过展示如何把这些概念翻译为 C# 语法，让你正确地开始掌握 C# 编程语言。这样，Beginning C# Objects 诞生了！

由于在我作为软件工程师的职业生涯中，一直关注 Java 技术，Apress 想找一些对 C# 较有经验的专业人士帮助我把本书从 Java “翻译”为 C#。Grant Palmer，本书的另一位作者，以及 James Huddleston，我的第一技术审校，都是极好的合作对象。我非常高兴能有机会与他们二位合作编写本书。

—Jacquie Barker

Introduction

介绍

本书最初和最重要的关注点，是软件对象：什么是软件对象，为什么它们如此神奇而直白，以及如何正确使用对象去构建一个软件应用。

这也是一本关于 C# 的书：它并非一本“C# 大全”，而是一份对该语言的温文但全面的介绍，还特别强调了如何把对象模型转换到功能完备的 C# 应用程序——其他图书很少这样编写。

本书的目标

我们撰写本书的目标（希望也是你买它的目的）是

- 让你熟悉基础的面向对象(OO)术语和概念。
- 让你得到对象建模领域的上手实践体验：即，创建一幅可用作后续面向对象软件系统构建基础的“蓝图”。
- 演示如何将类似对象模型转换为可用的软件应用程序，特别地，C# 应用程序（虽然你将学到的对象建模技术也可用于其他语言）。

如果你已经熟悉 C# 语言（但不熟悉对象基础），那么，学习 C# 的面向对象根源，对你成功使用该语言至关重要。另一方面，如果你是 C# 新手，本书将让你能正确地“起跳”。从这两方面看，对于任何打算成为精通一门类似 C# 的 OO 编程语言的人，本书都是必读之作。

本书并不打算做到。

- 让你一夜之间精通对象建模：和其他高级技能一样，完全掌握对象建模技术有两个条件：良好的理论基础和大量的练习！本书只告诉你基础知识，以及让你能应用和实践新学知识的项目/练习的建议。但是，真正精通对象建模技术的必经之路，还是坚持不断地参与 OO 建模和项目开发。本书给你以技能，希望还能给你以信心，让你能在一个专业机构中开始应用对象技术，那才是你真正学习的地方，尤其是如果拥有一位 OO 经验丰富的导师在整个“工业标准”项目中指导你时，更是如此。

- 让你变成某种特定 OO 术语体系的专家：OO 软件开发领域有数十种不同的正式方法，新变种还在不断产生，而且并无优劣之分。例如，UML（即 Unified Modeling Language，统一建模语言）符号体系是最新的，而 OMT（即 Object Modeling Technique，对象建模技术）是最旧的其中之一，然而两者极为相似，因为 UML 基本上极大地扩展了 OMT。在理解了对象建模技术的一般过程和用 UML 建模的技术后，你就拥有足够的知识，可以自己检阅、评估和选择一种特定的术语体系（或创建你自己的体系——甚至也许有一天你会写一本关于自己发明的术语体系的书籍，谁知道呢！）

- 教你你所需的全部 C# 知识：C# 是一种非常丰富的语言，在 Framework Class Library 中包括了数十个核心类、数百个其他类，以及这些类提供的数以千计的操作。假使 C# 提供了一打实现某种操作的方案，则我们将只介绍其中一两种最适合解决手边问题的方案，让你明白解决问题的方式。尽管如此，在本书中，你还是一定会看到足够用来构建一个完整应用的 C# 语言知识。

用从本书学到的基础知识武装好自己后，你将能欣赏关于该领域的更为彻底的阐述，例如坊间有售的 C# 参考书中的知识，或 UML 参考书中的描述。

对于想做成功 OO 程序员的人，为什么理解对象如此要紧

我们常常与一些软件开发会面（在工作场所，在客户办公室，在专业会议上，或在大学校园里），这些开发者都尝试去掌握一门类似 C# 的 OO 编程语言，他们参加 C# 培训，阅读关于 C# 的书，或是安装和使用像 Visual Studio .NET 这样的 C# 集成开发环境（IDE）。然而，这是舍本逐末的做法：他们缺乏对什么是对象的基础认识，更为严重的是，缺乏利用对象从头开始构建软件应用程序的知识。