C# 3.0 Design Patterns

针对C# 3.0的最新内容

C# 3.0 设计模式

O'REILLY®
机械工业出版社
China Machine Press
HZ BOOKS

(南非) *Judith Bishop* 著
王江平 译

C# 3.0 设计模式

(南非) *Judith Bishop* 著

王江平 译

Beijing • Cambridge • Taipei • Tokyo

机械工业出版社出版

机械工业出版社

图书在版编目（CIP）数据

C# 3.0 设计模式 /（南非）布什波（Bishop, J.）著；王江平译．－北京：机械工业出版社，2009.1

书名原文：C# 3.0 Design Patterns

ISBN 978-7-111-25080-7

I. C... II. ①布 ... ②王 ... III. C 语言－程序设计 IV. TP312

中国版本图书馆 CIP 数据核字（2008）第 140939 号

北京市版权局著作权合同登记

图字：01-2008-1734 号

书　　名 / C# 3.0 设计模式

书　　号 / ISBN 978-7-111-25080-7

责任编辑 / 陈佳媛

封面设计 / Karen Montgomery，张健

出版发行 / 机械工业出版社

地　　址 / 北京市西城区百万庄大街 22 号（邮政编码 100037）

印　　刷 / 北京牛山世兴印刷厂

开　　本 / 178 毫米 × 233 毫米　16 开本　19.5 印张

版　　次 / 2009 年 1 月第 1 版　2009 年 1 月第 1 次印刷

印　　数 / 0001-3000 册

定　　价 / 49.00 元（册）

凡购本书，如有倒页、脱页、缺页，由本社发行部调换。

本社购书热线：（010）68326294

O'Reilly Media, Inc. 介绍

为了满足读者对网络和软件技术知识的迫切需求，世界著名计算机图书出版机构O'Reilly Media, Inc. 授权机械工业出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc. 是世界上在 UNIX、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时也是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的50本书之一）到GNN（最早的Internet门户和商业网站），再到WebSite（第一个桌面PC的Web服务器软件），O'Reilly Media, Inc.一直处于Internet发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc. 是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc. 具有深厚的计算机专业背景，这使得O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc. 还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc. 依靠他们及时地推出图书。因为O'Reilly Media, Inc. 紧密地与计算机业界联系着，所以O'Reilly Media, Inc. 知道市场上真正需要什么图书。

译者序

这本书是基于3.0版的C#语言来讲解设计模式的。模式依然是那23个最经典的面向对象设计模式，即Gamma等人1995版《Design Patterns：Element of Reusable Object-Oriented Software》（编注1）一书中讲到的模式。但老瓶装上新酒，却也透出一些新味道，主要“新”在：

1. 正如书名所言，本书是基于C#语言的，本书所有的示例程序全部基于C#语言及.NET框架，这一点跟95版《设计模式》不同，在那本书里，示例代码以C++语言为主，辅以SmallTalk。许多C#程序员不熟悉那些语言，所以本书显然更适合C#程序员。
2. 本书在讲解设计模式的同时，也讲解了许多C#语言特性（有25种之多），特别是C# 3.0的新特性。这可以使读者在学习设计模式的过程中进一步加深对语言特性的理解。
3. 本书采用的示例代码基于C#语言及其新特性，而C#语言的许多高级特性为模式设计提供直接支持，比如yield return和IEnumerable接口直接支持迭代器（Iterator）模式，克隆和序列化则可简化备忘录（Memento）模式的设计。
4. 本书采用的示例是全新的，它们更接近当前软件行业的前沿。这些示例使本书对模式的讲解更加深入浅出。译者个人认为，在个别模式上本书的讲解不亚于95版《设计模式》，比如对解释器（Interpreter）和享元（Flyweight）模式的讲解。

虽然人们一般认为编程思想高于语言，但编程语言和设计模式却总是相互影响：一方面，合理地运用设计模式可以充分发挥出语言功能的强大威力；另一方面，语言对设计模式支持的程度越高，模式就更容易被正确使用和推广。本书的一大特色就是将这一理念渗透在全书中。结尾处还介绍了C#语言在这方面所做的努力以及未来的发展方向。

在翻译本书的过程中，译者根据作者在配套网站（*http://patterns.cs.up.ac.za/*）上的勘误进行了更正。

特别感谢来自CSDN的网友张磊（网名xuelong_zl－点雨点），他不辞辛苦，一字不漏地复审了全书的译稿，协助查出了初稿中不少的错别字、病句，并对原稿中许多晦

编注1：该书已由机械工业出版社引进出版：《设计模式：可复用面向对象软件的基础》（7-111-07575-7）。

涩的译句提出了修改意见。此外，还要感谢另一名来自CSDN的网友高远（网名：Wolf0403－完美废人），他协助推敲了个别疑难句子的翻译。正是有了以上两位的协助，本书的翻译质量才得到进一步的保障。

最后还要感谢我亲爱的妹妹，多少次敲打键盘到深夜，她总会倒一杯甘甜的果汁放在我电脑的旁边。

客套话就不说了，译者已使出浑身解数，有优点您多吹捧，有缺点您就狠狠地批吧。

王江平

2008年5月写于上海浦东

作者简介

Judith Bishop教授执教于南非比勒陀利亚大学计算机科学系。她专注于编程语言在分布式系统中的应用以及基于Web的技术。她作为新技术的倡导者而闻名于世，所著Java和C#方面的著作使用6种语言在全世界发行。Judith是IFIP（International Federation for Information Processing，国际信息处理联合会）TC2（软件）的南非代表，她还是多个国际会议委员会及编委会的主席或成员。

封面介绍

《C# 3.0设计模式》的封面动物是一只灰雁。灰雁大概是最早被人类驯化的动物之一。考古学研究证明，在3000年之前的古代埃及和罗马就已经出现了驯养的灰雁。

灰雁的个头比较大，体重通常在5-12英磅之间，双翼展开后平均宽度有59-66英寸，长度通常是29-38英寸。灰雁的羽毛是灰褐色的，腹部是白色的，胸的下部是暗灰色。它们的嘴巴是黄色的，而且比较大，腿和脚是粉红色的，有点像肉的颜色（幼雁的腿是灰色的，而脚随着年龄的增长越来越红）。

灰雁是候鸟，它们会在冬天飞到南方或者西方来躲避严寒天气。在夏天，他们会居住在苏格兰、冰岛、斯堪的纳维亚，往远东方向一直到俄罗斯，以及波兰和德国。在秋天，冰岛的灰雁会迁徙到不列颠群岛，而欧洲其他地方的灰雁则会迁徙到荷兰、西班牙、法国和东非。

灰雁是一种群居鸟类，它们总是结伴成群做长途旅行，经常排成人们所熟悉的V字形。雁群中雁的数量也有多有少，小的群是只是个小小的家族，大的群中可能聚集成千上万只雁。

灰雁的繁殖季节随地理位置而不同。在苏格兰，繁殖期从四月下旬开始；冰岛则开始于五月上旬；在欧洲则开始得更早一些。在繁殖季节，灰雁居住在沼泽地中——植被繁茂的地方。雁巢筑在高处，以保护它们的卵不被掠食者吃掉。

母雁最多能产12只卵，但通常是4到6只。孵化时间大概是26天。出生后，雏雁要等到身体干燥之后才能离巢。幼雁在父母的指导下开始学着喂养自己。它们的平均寿命是20年。

灰雁以沼泽地上的草、根和块茎为食，也吃小的水生动物。它们还喜欢吃一些根块农作物——比如萝卜、马铃薯和胡萝卜——这一点就关系到欧洲农民的切身利益了。

雕类、乌鸦和鹰是灰雁的空中掠食者；在陆地上，它们要警惕野狗、狐狸和人类。人类猎取灰雁是为了享用它们美味的肉和柔软的绒毛。它们的绒毛常被用来填塞枕头、毛毯及户外服饰。罗马皇帝凯撒曾在公元前390年将灰雁奉为神灵，禁止杀戮和享用它们。

凯撒相信灰雁可以保护其帝国不受外来侵袭。他相信当高卢企图入侵时，是大雁响亮的叫声为罗马人拉响了警报，从而使他们的国家免于被占领。

目录

序言

当你面对一个需要解决的问题时（坦白地说，谁没面对过呢），我们搞计算机的人通常采用的基本策略就是“分而治之”（divide and conquer）。它的基本过程是这样的：

- 将待解决问题概念化，将其划分成一系列更小的子问题；
- 解决每一个子问题；
- 将结果组合起来，使其成为原问题的解决方案。

将复杂的问题逐步分解，一直分解到只需操作几十亿比特的状态值，这是我们每天都在做的事情。但是，“分而治之”并非惟一可行的策略。我们还可以采取更加泛化的方法：

- 将待解决的问题概念化，使其成为某个更一般的问题的特例；
- 采用某种方式来解决那个更一般的问题；
- 将一般问题的解决方案做适当调整，以解决这个特殊问题。

对喜欢泛化方法的人来说，设计模式是他们的主要工具。如果你能从大量的、不同领域的软件解决方案中挑出一些例子进行比较，你就会发现，尽管涉及的业务细节千差万别，它们却往往有着相似的基础结构（从某种意义上，以下两种行为的结构是类似的：1.在文件系统中查找拥有特定属性的文件，2.从分析树上查找特定类型的符号）。设计模式就是从这些一般问题的通用解法中整理出来的。

将泛化方法运用到极致的例子莫过于编程语言本身的设计与实现。就解决问题的工具而言，很难找到比C#语言更通用的了。当设计新的编程语言（或者旧编程语言的新版本）时，我们会考虑开发者们每天都要面对的一般问题，确定如何创建出一种新的语言，能够以通用、优雅、强大的方式来解决它们，并且提供广泛的可用性。

我们想把那些最有用、最强大的抽象机制深深植入到语言的基础设施中，使你的意识里甚至忘记了这些抽象的存在。诸如“局部变量”、“过程调用”和“while循环”这样的模式已经成为我们所呼吸的空气的一部分，以致于我们现在都不觉得它们是模式了。

进一步讲，我们想使那些有用，但并不那么基础的模式，在我们的语言中也能以相对直接的方式来实现，并且不失清晰和优雅。C#中的类可标记为“static”，“abstract”或“sealed”，但不可以标记为“singleton”。这是语言设计者们的有意安排。尽管如此，用C#实现一个单例模式还是相对容易的。

在“必要基础”和“偶尔有用”之间的区域就是考验设计的有趣地带了。我们观察了实际开发者在C#（及其他语言中）对设计模式的使用，而这有力地驱动了我们新版本语言的设计过程。

考虑这样一个例子：在C# 1.0中，你怎样在链表上实现迭代器（Iterator）模式。你可能会定义一个枚举器类型来表示一个链表中的位置，这个链表类中包含大量令人厌烦的、影响可读性的样板代码，这样一个方案的可复用性并不怎么好。“枚举集合中的东西”，这种观念在大量的问题中都非常适用，所以有必要把它包含进语言中，成为语言概念中的一等公民。在C#2.0中，通过使用yield return语句，所有那些烦冗的代码都可以由编译器帮你产生出来，而泛型则为集合元素的迭代提供了类型安全性，而不管集合中的“东西”到底是什么。

所有这些长篇大论都解释了为什么我对C# 3.0的语言级集成查询（Language Integrated Query，LINQ）感到如此激动。我们相信，迭代集合中的东西是一个意义重大的开端，但我们还可以做更多。数据的排序、过滤、分组、连接、投影和转换也都是很多领域中的基本操作。不管你是在实现一个射线追踪器、编译器、XML阅读器或在线银行安全系统，你都可能通过多种方式操纵元素的集合。

将这些概念从领域特定的对象模型中提取出来并融入到通用的编程语言中，我们就有希望解决更一般的问题。然而，我们更希望通过C# 1.0、2.0中已有的丰富特性，再

加上 C# 3.0 中的查询表达式、lambda 表达式、扩展方法、初始化表达式、表达式树等，可以更加轻松地实现其他各种有用的设计模式，而且不失优雅。

而这也是我为本书感到激动的原因。本书为那些充满奥秘的设计模式提供了实效性的 C# 3.0 实现。我非常希望了解开发者能在何种场合利用好这些工具和这门语言，以及哪种有用的模式可能融入到将来语言的基础设施中。

——C# 编译器开发组资深开发者 Eric Lippert

2007 年 11 月 30 日于华盛顿州西雅图市

前言

我为什么写这本书

2002年，微软研究院（Microsoft Research）在英国的剑桥召开了一次国际会议。在这次会议上，微软展示了它的Rotor系统，这个系统将C#和.NET带给了非Windows程序员。会议结束之后，我回到家开始编写相关软件、论文和书，这一过程中我意识到：我们已经见证了一轮真正的程序设计革命的开始。自1996年Java出现以来，程序设计变得平台独立：通过Java的字节码，程序可以到处运行。然而，这种独立性只适用于Java编写的程序。与此不同的是，.NET是独立于语言的：它允许不同语言编写的程序彼此交互，只是在那一天到来之前，这仅限于Windows系统。

在后面的五年里，出现了新的支持.NET的平台（比如Mono）以及新的支持Intel芯片的硬件（Windows跑在Intel芯片上）。结果就是，如今.NET几乎也可以到处运行了。于是，C#程序设计也成了一门可移植的编程技术。但作为一门语言，C#一直在改进，现在我们处在了向3.0飞跃的起始点上。为带来更高的生产率和更好的易用性，C# 3.0提供了大量有用的新特性。我在2003年写过一本介绍C#语言的书（注1），4年后（2007年）发布的新特性所带来的好处，使我觉得C# 3.0将软件开发推向了一个更高的层次。我曾经想再写一本书，把C# 3.0介绍给那些已经掌握语言基础的开发者，但采用何种的形式才能更好地介绍一种编程语言，同时满足读者在精确性、示例方面的要求以及大量的现实性需求呢？

我想到了设计模式。设计模式封装了综合运用语言特性来解决问题的一般的、公认的和有效的方法。它们提升了讨论的层次，实现并发展了好的编程方法。然而，设计模式总

注1：书名为*C# Concisely*由Judith Bishop和Nigel Horspool合著于2003年由Addison-Wesley出版。

带给人们一些不现实的感觉，人们对设计模式的印象常常是：讨论得多，用得少。我想改变这一点，借助于为其准备的最好的编程语言：C# 3.0，让普通的开发者也真正理解设计模式。

本书为谁而写

如果你是一个钟爱自己代码的程序员，想让代码的每一行都有准确的含义，每一个特性都用在正确的场合中，这本书就是为你准备的。它将帮助你完成最基本的工作：使你的代码正确、优雅、可扩展而且高效。如果你在公司里负责代码质量，你也需要一本这样的书。对于那些致力于成为软件工程师或架构设计师的低层程序开发者，掌握设计模式的知识也是一大进步。

通过阅读本书，你可以掌握以下技能：

- 设计模式编程；
- 用来描述模式的基本 UML 建模表示法；
- 选择适合于特定情形的模式，并比较不同的实现方式；
- 使用 C# 3.0 的高级语言特性来高效、优雅地实现模式。

尽管本书并没有写成一本教科书，它完全可以用于设计模式和高级程序设计方面的中级课程。

本书中所有模式及相关示例的图、代码和研究案例都可以从本书网站下载：*http://www.oreilly.com/catalog/9780596527730*。

阅读本书需要的知识

本书是为这种程序员准备的：至少了解 C# 1.0 或 Java1.4 程序设计并希望进一步转到最新版本的语言上，使用更加现代化的特性进行程序开发。C# 3.0 的所有新特性，以及许多来自于 C# 2.0 的较新颖的特性，在本书中都通过示例做了介绍，为方便引用，本书还在注释文字中进行了总结。因此，本书也可以作为语言的使用指南。

本书并不要求读者熟悉设计模式。它已涵盖了全部 23 种经典模式。这 23 种模式是 Erich Gamma、Richard Helm、Ralph Johnson 和 John Vlissides 四人于 1994 年在他们的《Design Patterns: Elements of Reusable Object-Oriented Software》（中文书名为《设计模式：可复用面向对象软件的基础》）中提出的，而且现在已成为其他领域（如安全、并发和框架程序设计）中出现的新模式的基础。看完本书，读者将深入掌握设计模式基础，就像大多数人所理解的那样。