第一图书网, tushu007.com

<<C#3.0设计模式>>

图书基本信息

书名：<<C#3.0设计模式>>

13位ISBN编号：9787111250807

10位ISBN编号：711125080X

出版时间：2009-1

出版时间：机械工业出版社

作者：布什波

页数：294

译者：王江平

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

前言

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。

内容概要

如果想加速．NET应用程序的开发，你就需要本书，它是解决常见编程问题的优雅的、公认的和被证明为有效的处理方式的合集。
本书条理清晰地介绍了经典的面向对象设计模式，并解释了怎样运用最新的C#3．0语言特性来编写它们。
C#设计模式利用新的C#3．0语言和．NET 3．5框架中的特性来实现开发社区中广为人知的23种设计模式。
本书包含了揭示每种模式在实际中如何运用的丰富案例，以及对于模式及其最佳使用场合和最佳组合方式的富有洞察力的比较。
本书结构清晰、示例丰富，涵盖以下内容：　　· 介绍设计模式以及为什么使用它们，其中的表格和使用指导可以帮助你从多种模式中选出想要 的模式。
· 各种经典设计模式的演示，包括创建型模式、结构型模式和行为型模式，还有每种模式的UML表示以及其中各参与者所扮演的角色。
· 从示例中引入C#3．0语言特性，并在文本框中对它们进行总结，以方便引用。
· 每个模式的示例都是实际可运行的．NET 3．5程序，可以从O ’ Reilly网站和作者的配套网站上下 载。
· 检验你是否理解书中内容的小测试和练习。
在本书中，你将学习如何编写正确、可扩展和高效的代码，以节省前期的开发时间，消除后期的软件问题。

作者简介

Judith Bishop是南非比勒陀利亚大学计算机科学系教授。
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。
她是新技术的倡导者，所著Java和C#方面的书籍以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。

书籍目录

序言前言第1章 C#与设计模式关于模式关于UML关于C#3．O关于示例第2章 结构型模式之装饰器、代理与桥接装饰器模式代理模式桥接模式示例：OpenBook模式比较第3章 结构型模式之组合与享元组合模式享元模式模式比较第4章 结构型模式之适配器与外观适配器模式外观模式模式比较第5章 创建型模式之原型、工厂方法与单例原型模式工厂方法模式单例模式模式比较第6章 创建型模式之抽象工厂与生成器抽象工厂模式生成器模式模式比较第7章 行为型模式之策略、状态与模板方法策略模式状态模式模板方法模式模式比较第8章 行为型模式之职责链与命令职责链模式命令模式模式比较第9章 行为型模式之迭代器、中介与观察者迭代器模式中介模式观察者模式模式讨论和比较第10章 行为型模式之访问者、解释器与备忘录访问者模式解释器模式备忘录模式模式比较第11章 设计模式展望模式总结设计模式展望结束语附录参考书目

编辑推荐

本书结构清晰、示例丰富，涵盖以下内容：介绍设计模式以及为什么使用它们，其中的表格和使用指导可以帮助你从多种模式中选出想要 的模式。
各种经典设计模式的演示，包括创建型模式、结构型模式和行为型模式，还有每种模式的UML表示以及其中各参与者所扮演的角色。
从示例中引入C#3．0语言特性，并在文本框中对它们进行总结，以方便引用。
每个模式的示例都是实际可运行的．NET 3．5程序，可以从Oreilly网站和作者的配套网站上下 载。
检验你是否理解书中内容的小测试和练习。
在《C#3.0设计模式》中，你将学习如何编写正确、可扩展和高效的代码，以节省前期的开发时间，消除后期的软件问题。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com