<<设计模式>>

图书基本信息

书名：<<设计模式>>

13位ISBN编号：9787121075070

10位ISBN编号：7121075075

出版时间：2009-1

出版时间：电子工业出版社

作者：王翔

页数：652

前言

如果你要开发一个小型的系统，整个系统只有两三人，系统活不过五年，商业逻辑单纯，程序代码不超过万行，那么你随便做，影响不大。
反之，若要架构大型系统，你需要慎思，套用模型与架构，将前人的经验当作基石，这样系统设计才不至于陷入发散。

然而，你不会一开始就做大系统，那样风险太高。
因此，要练习，最好从小系统开始使用模型与架构，这样才能检讨与学习，日后方能在大系统中自如运用。

企业信息系统现今面临着大量的整合需求，需要提供深入的分析应用和灵活的应变流程。
但系统整合的复杂度是彼此系统复杂度的乘积，系统间的安全、弹性、效率、扩充性、可用性……彼此互相掣肘，此时，企业需要接触广、想得深、能定方向的架构师。
而熟悉设计模式是架构师的养成基础，要求对于问题的分类与解法有一定的认知。

有经验的设计者们，抽象出系统开发的原则与标准问题的设计解法，而GOF于十几年前提出的23种模式是其中的佼佼者。
但毕竟空有概念，仍难落实到你日常使用的程序语言中来。
坊间许多图书作者利用不同的程序语言，例如C++、Vistlal Basic、.NET、Java等，实现GOF的23种模式，配合UML的模型说明，让你可以方便地应用在自己的开发环境中。

本书的作者王翔有多年的开发经验，参与过多个千万乃至十亿行代码的大工程，他将经验融于设计模式中，以c}}重新实现了GOF的模式，同时加入了新近的设计想法，如SOA与Web Setvices等，以及相对于其他设计模式而言较新的.NET Framework实现技术，如泛型、3.0的WCF等。
在本书中他除了正向地以C#展现多个不同用途的模式外，还提供了日后可重复验证与测试的单元测试码。

系统分析与设计是门艺术，问题的解法与何为问题是交织的，而各模式的搭配使用技巧不同，要领存乎一心，须要巧思与反复琢磨，方可有好的解法。
本书立意明确，除了告诉你问题的类型与解法，还提供了可以立即演绎的程序代码。
相信这本案头的工具书可以提供你一个不错的思维模式，帮你造就有弹性、能扩充、易维护的软件实体。

须要提醒你的是，抽象化的思考、封装与重用的设计神髓在心中，而不是落在纸上的程序代码，阅读此书时，不要停止在仅Copy and Paste程序代码。

内容概要

本书基于C# 2.0的语法，试图将GOF 23中的模式以一种可工程化的公共库而非Example的方式呈现给读者。

内容包括以下7部分。

第1篇主要是概括性的介绍；第2篇创建型模式介绍通过将实例化职责委托他方对象的办法，隔离客户程序与具体类型实例化的依赖关系，保证客户程序（或者外部系统）获得期望具体类型实例的、同时不必发生直接的引用；第3篇结构型模式的重点在于如何通过灵活的体系组织不同的对象，并在此基础上完成更为复杂的类型（或者类型系统），而参与组合的各类型之间始终保持尽量松散的结构关系；第4篇行为型模式关注于应用运行过程中算法的提供和通信关系的梳理；第5篇主要介绍小颗粒度基础模式和应用案例；第6篇主要介绍应用全局的模式化的实现方法，包括现在已经被普遍应用的N层模式及某些关键性框架产品采用的“微内核”模式；第7篇主要是一些针对Web和Web Service领域的模式设计技术。

本书主要针对对C#语言和.NET Framework平台有一定了解或有一定应用经验的用户，尤其适于那些希望运用模式技术在设计和开发方面多应对些挑战的用户。

作者简介

王翔，软件架构师，主要从事.NET、XML、公钥基础设施的开发。
专注于数据（尤其是XML信息）的生产、加工、交换、提炼等过程。
最近参与了一系列有关应用密码技术和PKI环境保护信息系统数据安全的项目。
最喜欢数学，平常案头总是摆一本数学练习题。
闲暇时间喜欢写作，通过发表多种技术文章与国内外同行交流各种数据应用经验。
项目间隙经常到各海滨城市徒步旅行、野外露营、出海航行、极限运动，这几年烹饪也渐渐成为个人主要爱好。

<<设计模式>>

书籍目录

章节摘录

第1篇　预备知识——发掘用C#语言进行面向对象化设计的潜力　第1章　重新研读C#语言　1.1　说明　本章以工程化使用为目的，对C#和.NET Framework提供的几个平时开发时不引人注意的特征进行介绍，它们对于提高代码扩展性、灵活性很关键，务求用“很C#”的方式解决以往设计模式Example之外必须面临的一些问题。
它们主要包括：　·Namespace（命名空间）；　·Delegate（委托）；　·Generics（泛型）；　·Attribute（属性）；　·Indexer（索引器）；　·Iterator（迭代器）；　·Overloadable Operators（可重载运算符）与Conversion Operators（转换运算符）；　·Configuration（对象化配置访问）；　由于降低类间耦合关系一直是设计上控制变化范围的常用手段之一，所以在设计模式之外补充有关用C#实现“依赖注入”（Dependency Injection）的方法。
有关设计模式在Threading（多线程）模型下实现需注意的内容，笔者将在相关章节的工程化分析部分介绍。
1.2　C# 部分语法内容扩展　1.2.1　命名空间（Namespace）　尴尬的现实状况　是否有很好的命名空间规划是工程化代码与非工程化代码一个很明显的区别。
尤其对于大型的组织而言，如果涉及的产品线、项目、公共平台很多，如何通过命名空间把所有的代码资源有效地组织起来，恐怕是实施项目前要考虑的主要问题。
作为一个树形体系，最好有组织级统一的分类标准，目的很明确——用的时候能很容易找到。
做到这点并不容易，原因如下：　·习惯中很难改变的缩写命名：CAD很容易在声明的时候被命名为.***CAD，但事实上Design Guideline建议的是Cad，这样使用者和定义者之间就存在一些小小的错位。
在99%的情况下这个问题不会发生，因为您只要一个点，IntelliSense就帮您列出来了，另外的1%则发生在后绑定调用的情况下。
·不统一的命名：涉及加密的库，可能A被定义为Encrypt，B被定义为Crypto或Cryptography。
来自匈牙利命名法的“遗毒”也常常成为新旧开发人员统一命名的障碍，而且会直接影响到命名空间的定义。
·组织或您上司的认同：技术总监不认为命名规范是他需要关心的事情，下面的架构师更关心的是结构，再下面的项目经理关注的是资源调度和进度，具体的实施人员恐怕没有多少机会规定其他同事该怎么命名，那么谁来关心命名空间呢？
企业.NET类型系统的命名空间规划示例　无论如何，即便没有办法在组织级统一命名空间，为了您所带领的团队现在做的工作在以后能更容易地被应用，或者仅仅为你自己的职业生涯好好“储蓄”，在动笔编写第一行程序之前，先规划好命名空间吧。

媒体关注与评论

本书立意明确，除了告诉你问题的类型与解法，还提供了可以立即演绎的程序代码，相信这本案头的工具书可以提供你一个不错的思维模式，帮你造就有弹性、能扩充、易维护的软件实体。

胡百敬　微软MVP，台湾恒逸资讯资深讲师，“数据库铁人”　　作者从GOF23种经典设计模式开始，带你走进模式的失门，小到细粒度的基础模式，大到粗粒度的架构模式，本书都做了详尽的讲解。

如果您还在为了软件需求的无尽变化而烦恼不断，为了在软件设计领域更上一层楼而苦苦思索，希望本书能够带给您一些启发。

李会军　微软MVP，博客园专家，IT168专栏作者　　本书很有特色的地方，就是以工程角度来阐释模式，相较纯粹的模式之说，则更具普遍的下手角度，C＃语言的高级特性结合设计模式的经典思想，两者相得益彰。

王涛　微软MVP，博客园专家，《你必须知道的.NET》作者

编辑推荐

专家推荐　　本书立意明确，除了告诉你问题的类型与解法，还提供了可以立即演绎的程序代码，相信这本案头的工具书可以提供你一个不错的思维模式，帮你造就有弹性、能扩充、易维护的软件实体。

胡百敬　微软MVP，台湾恒逸资讯资深讲师，“数据库铁人”　　作者从GOF23种经典设计模式开始，带你走进模式的失门，小到细粒度的基础模式，大到粗粒度的架构模式，本书都做了详尽的讲解。

如果您还在为了软件需求的无尽变化而烦恼不断，为了在软件设计领域更上一层楼而苦苦思索，希望本书能够带给您一些启发。

李会军　微软MVP，博客园专家，IT168专栏作者　　本书很有特色的地方，就是以工程角度来阐释模式，相较纯粹的模式之说，则更具普遍的下手角度，C＃语言的高级特性结合设计模式的经典思想，两者相得益彰。

王涛　微软MVP，博客园专家，《你必须知道的.NET》作者

版权说明