COODER 程序员诚品

没有谈神马，也没有讲浮云

漫谈设计模式，也漫谈面向对象（OO）

放弃过程式编程（FP/PP），玩转面向对象编程（OOP），

辅以面向切面编程（AOP）

不要if-else，不要重复自己（DRY）

反转控制（IoC），反转依赖（DI）

单例模式，工厂方法模式，

装饰器模式……

原型模式，爱编程，也爱琢磨

爱钻研，更爱分享

我是程序员

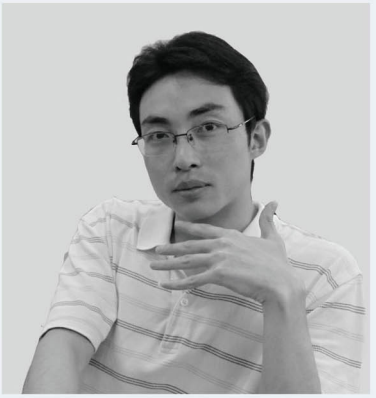
漫谈设计模式背后的软件设计思想

刘济华 编著

漫谈设计模式

——从面向对象开始

清华大学出版社



喜欢漂亮代码，
更喜欢写漂亮代码
爱听音乐，爱看电影，
也爱打魔兽
喜欢技术，喜欢艺术，
也喜欢黑格尔　我不是大虾

I'm a loud-mouth on software development

刘济华 网名Co0der，现任惠普公司ITSM顾问，曾在电信、金融、保险等多个行业参与开发了多个大型项目和产品，参与过UMS系统、保险核心业务系统、银行系统、CRM系统、ITSM系统软件研发。热衷于编写优良代码以及研究如何提高开发效率，对面向对象开发与设计有较深层次理解，致力于使用OOP为复杂领域问题建模。喜欢钻研并应用一些出色的模式，以及支持软件高效开发设计的流程和方法，是领域驱动开发（Domain-Driven Design）和敏捷（Agile）开发方法的忠实粉丝。

个人博客：http://redhat.iteye.com/

内 容 简 介

模式引入计算机科学领域已经有 20 余年了，最初人们侧重于面向对象的设计，而现在已经应用于计算机领域的各个方面，它们对计算机产生了深远影响。

若想一本书涵盖所有模式，那么这本书将会非常庞大，以前的设计模式书籍专注于介绍设计模式，虽然读者了解了这些设计模式，但是仍然不知道如何合理地使用它们，往往导致读者为了使用设计模式而设计，而不是从问题出发，使用设计模式优雅地解决这些问题。

本书主要从最基本的设计模式入手，并结合一些 J2EE 开发过程经常遇见的技术和概念，你将全面理解这 10 多个设计模式，并在开发过程中，让你真正体会和思考面向对象编程的思想，也只有掌握这些，你才会能成为一位真正的设计专家。

本书不是一本面向对象和 Java 语言的入门书籍，阅读对象主要是从事 Java 语言的软件开发人员但不限于 Java 语言的开发人员。

图书在版编目(CIP)数据

漫谈设计模式——从面向对象开始 / 刘济华 编著. --北京：清华大学出版社，2012.1

ISBN 978-7-302-27302-8

Ⅰ. ①漫… Ⅱ. ①刘… Ⅲ. ①JAVA 语言—程序设计 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2011）第 223939 号

责任编辑：栾大成
装帧设计：
责任校对：徐俊伟
责任印制：

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：010-62795954, jsjjc@tup.tsinghua.edu.cn
质量反馈：010-62772015, zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：188×230　　**印　张**：18.5　　**插　页**：1　　**字　数**：358 千字
版　　次：2012 年 1 月第 1 版　　**印　次**：2012 年 1 月第 1 次印刷
印　　数：1~5000
定　　价：45.00 元

产品编号：043695-01

目 录

第一篇 模式介绍

第二篇 创建对象

第三篇 构建复杂结构

第四篇 行为模式

第五篇　终点还是起点

前 言

OOP（Object-Oriented Programming）早已不是一个新概念了，OOP 在最近的 20 多年里发展得异常迅猛，特别是近 10 年里，OOP 相关技术层出不穷，当大家热衷于使用这些新技术时，却不会使用 OOP 进行软件设计，进而发现新的技术并没有为大家带来任何好处。

很多老的开发人员从**过程式开发**转向**面向对象的开发过程**中，由于他们习惯过程式思维的开发，尽管他们使用的是 OOP 语言，但这并没有给他们带来太多帮助，反而使他们更加厌倦 OOP 软件开发，认为 OOP 没有想象中的那么便捷，很多地方没有使用过程式开发来的便捷，于是他们又退化为过程式的开发。

随着越来越多的新开发人员也加入了 OOP 的潮流，他们追求新的技术，学会使用各种工具和框架，却无暇顾及 OOP 进行开发设计的核心思想。虽然使用了新技术，代码质量并未提高，反而事与愿违。

当他们沉浸在新技术的使用和业务逻辑的编码实现时，未料到这些看似巧妙的设计导致了代码不易阅读、不易维护、不易扩展、不易测试、不易调试……。大家忙忙碌碌，但是项目进度缓慢，最终往往以失败而告终。归根结底，尽管使用了 OOP 语言开发其他一些新技术，但对 OOP 只限于粗浅的了解和相关语言语法使用上的理解，并不会真正使用 OOP 进行开发设计，以致使用时颠三倒四，未能真正享受到 OOP 和这些新技术带来的好处，有些新技术非但没有提供帮助，反而成为某些软件失败的罪魁祸首。那么，如何使用 OOP 进行开发设计呢？

OOP 开发新手由于没有这方面的设计经验，在遇到问题时，往往诉求于逻辑的实现，在维护性和扩展性没有考虑或者少有考虑，导致代码乱七八糟，七零八散，随着开发的深入，最终在用户各种各样的需求面前无以应对。而有经验的 OOP 开发人员会灵活使用各种模式做出优秀的设计，编写的代码健壮性高，易于阅读、维护和扩展，可伸缩性强，开发成本也十分低廉。如果重用他们的开发经验，那么你就不需要在相同问题上重蹈覆辙，也能设计出优秀的软件。

市面上介绍设计模式的书籍非常多，它们一般仅仅给出 GoF 的 23 个最基本的设计模式的定义和一些简单的示例，大多数读者充其量只能了解它们，在使用上大打折扣。

本书精心筛选出了一些我们经常在开发设计过程中使用到的模式，使用 OOP 的眼光分析它们，适时结合一些流行 J2EE 框架和技术，并从横向和纵向两方面扩展读者的思维，使读者对这些常用的模式有一个全面深刻的认识，也希望能够为正在使用这些框架和技术的读者带来帮助。

本书内容

作为一本技术书籍，严谨是必须的，作者参考了大量资料和文献，并在相关重要地方注明出处，做到有理有据，希望能够从专业角度和用户探讨设计模式与面向对象的设计。

严谨和可读性是没有冲突的，在开发人员之间交流，最好的方式莫过于优质代码了，本书给出了大量代码片段，在一些重要的地方使用黑体、加粗或斜体的字体，并做了详细解释，希望能够抛砖引玉，帮助读者制作出更加出色的代码。另外，本书还添加了很多图片，希望图文并茂，使这本书更加容易阅读。

本书主要分为五篇。

第一篇：模式介绍（第 1 章～第 2 章）

第 1 章讲述了面向对象与模式之间的关系和模式的简史；在第 2 章介绍了第一个简单的模式，**模板方法模式，**在这章，我们分析了代码重复所导致的“腐臭气味”，重复的代码是代码“臭味”中最糟糕的，在以后章节将会介绍各种模式来避免代码重复。

第二篇：创建对象（第 3 章～第 6 章）

使用 OOP 语言的语法创建一个对象并不复杂，例如在 Java 语言中使用 new 即可。但是随着系统变得越来越复杂，使用 new 直接创建对象会给系统造成很高的耦合度。创建模式可以封装对象实例化的过程，把使用对象的功能和实例化对象解耦开来，从而降低了耦合度。

在本篇最后，讨论了现在最流行的两个概念，**IoC** 和 **DI**，这二者是目前流行的轻量级容器的基础。

第三篇：构建复杂结构（第 7 章～第 11 章）

有时候，创建新的、更强功能的类并不需要重新编写代码，装配已有的类和对象反而更加快捷，也更加灵活。本篇讨论了一些常用的组装对象的模式，你将发现，构建大

的、功能更强的对象，不是只有多层继承才能实现，组合往往是最有效的方式，本篇将会看到如何使用继承和组合创建复杂的大结构。

第四篇：行为模式（第 12 章～第 14 章）

我们在程序中经常需要封装一些对象的行为或者对象之间的通信，本篇将会讲述三个常用的行为模式：**策略模式**，**状态模式**和**观察者模式**，加上第 2 章介绍的**模板方法模式**，本书将一共讲述这四个常用的行为模式。

第五篇：终点还是起点（第 15 章～第 17 章）

其实，在很早之前，就有人对面向对象思想的局限性进行了研究，提出了一种新的编程方法 AOP，第 15 章将会介绍 AOP 的概念及其实现技术。

尽管在前面章节，我们学习了一些常用的模式和一些 OO 设计的原则，并且了解了 AOP 方面的一些知识，但这些并不能表示我们已具备解决复杂领域问题的能力，在第 16 章我们讨论了在实际开发过程中如何使用面向对象进行开发设计以及应该注意的一些问题。

在本篇末尾，第 17 章，将回顾本书的内容，重新认识 OOP 设计规范。

最后两章虽然没有直接讲解模式，但却是本书极为重要的部分，开始带领大家思考如何进行软件设计。

附录

本书附录 A 介绍了一些作者非常喜欢且推荐的书籍，以及一些经常浏览的技术网站和论坛，希望读者也能够获益。

参考文献。

本书读者

本书不是一本面向对象和 Java 语言的入门书籍，阅读对象主要是从事 Java 语言的软件开发人员但不限于 Java 语言的开发人员。

希望读者了解这些 Java 基本技术：反射、内部类、序列化、线程、动态代理、垃圾回收、类加载器以及 Java 对象的引用类型等。

本书中会使用到 UML 的一些图示，主要包括静态类图和时序图等。

希望读者了解或者使用过这些框架和技术：JDBC、Hibernate、Jpa、Spring、Ejb、Pico Container、Guice、XWork、Webwork、Struts 和 EasyMock 等。由于在实际的 J2EE 开发过程中，我们经常使用到这些框架，笔者将在讲述过程中适时结合这些框架与技术，用以消除因不理解这些技术而造成对模式理解上产生问题。如果读者还想做进一步的了解，可以参看我给大家推荐的相关书籍和网站。

对于其他语言学习者，由于文中有详细的代码解释，只要有 OOP 语言基础，相信能够看懂示例代码，可以重点关注于对问题的分析和 OO 眼光来看待这些设计模式。

对于 Java 初学者，对于提到的相关框架和技术，可以根据作者的介绍说明，重点专注于对问题的分析和 OO 眼光来看待这些设计模式，由于根据 Java 语法和技术的特点，扩展了一些模式，初学者可以根据作者介绍，更深入地体会这些语法和技术。

对于使用 Java 开发多年，却涉及 J2EE 方面的技术不深的读者来说，同样，重点专注于对问题的分析和 OO 眼光来看待这些设计模式，由于作者举了一些关于 Java 新语法和流行的技术，扩展了这些模式，可以作为参考学习，从多个角度扩展视野。

对于在 J2EE 开发多年的程序员来说，依然，你的重点还是要放在对问题的分析和 OO 眼光来看待这些设计模式之上，为了给读者广阔的视角，作者引用了大量流行的框架和技术，让读者能够理解这些框架和技术，有些示例可以直接拿过来解决这些常见的问题。

本书代码

要获取本书的示例代码，请登录 http://code.google.com/p/rambling-on-design-patterns/，点击进入 Downloads 标签页，选择最新的版本下载。也可以安装 svn 客户端下载最新文件，svn 地址为： http://rambling-on- design-patterns.googlecode.com/svn/trunk/。

为了能够让读者选择学习自己喜欢的章节并分别单独运行这些示例，作者尽量为每一种模式提供了相对独立和完整的代码。

> 注意：这些示例代码的运行环境是 Java 2 Platform Standard Edition SDK V6.0 及以上版本，代码使用了一些 Java 5 及以上的新语法和 JDK 的最新 API，如果读者对它们还不熟悉，请查阅附录的相关推荐书籍。

反馈

尽管笔者尽最大努力去避免正文和代码中出现的任何错误，但是人无完人，书中难免有错误之处。如果读者在阅读过程中发现拼写错误，代码错误，以及内容有混淆之处，希望能够及时反馈，或许它们能够节省其他读者很多宝贵时间，也有利于完善本书，反馈邮箱是：ramblingondesignpatterns@gmail.com。

写后感

为了完成此书，笔者增删了 5 次，尽管艰辛，但在写书的过程中也发现了不少乐趣，也希望为读者在阅读过程中带来乐趣。

好友赠言

书如其人

王林
惠普 ITSM 技术顾问

Co0der 具有典型的高级程序员的一切特质，在日常交流中，有时候让你很头痛，因为他常常会跟你较劲，你很难在一个论点上驳倒他。我细细阅读过他的书后，只有一个感觉，书如其人。这本书是 Co0der 对 Java 开发设计的总结，对目前仍旧在 IT 行业中作为底层“民工”而挣扎着的弟兄们有着极大的指导作用，能让这群人在技术上有一次不小的提高从而大有机会脱离“IT 民工”的范围。总而言之，试着去阅读吧，效果一定比你想象的要好。

一个痴迷于代码的程序员

风云
CENO 公司核心技术总监

刘济华在我的印象里是个痴迷于代码的人，大部分空闲时间都用来读书和钻研，博览群书、涉猎甚广。经年的沉淀终于汇总成书，这本书的最大优点是——使用了程序员喜爱的编码方式（请在书中认真体会“喜爱的编码方式”）解释概念，贯穿了很多流行的技术，并进行广泛的类比，讲述严谨。

总结的力量

李楠
普华永道资深软件工程师

这本书我读了一遍，感觉很不错。之前也看过一些设计模式书，经常觉得千篇一律，浮于表面，这本书完全是作者根据自己的心得体会写出来的——把复杂的事情变简单，通俗易懂，很有功力。读这本书的过程，能够体会到作者解决问题的思路，与自己的想法进行比较之后，使之前萦绕在心里的很多疑惑得到解决，而且书中的例子都可以独立执行，可以借鉴来在工作中直接使用。那些对设计模式不得要领的朋友可以看一下，是本值得一读的好书。

博采众家之长并加以提高

唐宇
惠普资深软件工程师

作者是我在惠普同一个研发项目组的同事，他是我认识的众多研发团队成员中很突出的一位，对技术的热爱和对完美的执着加上对知识的渴望就是我对他的印象。这本书给我的感览，是一本非常实用的设计模式书籍，它采用“娓娓道来”的方式，在大量国内外专家思想的精髓基础上，融入了作者独到的理解，再根据中国人的学习习惯加以归类和总结。总的来说，我想推荐这本书给学习设计模式的专业编程人员。

不打不相识

钟俊华 (Peyton Zhong)
易保资深软件工程师

了解济华是在跟他的一次激烈争论之后，记得是一个关于设计的问题，那次争论之后我才真正体会到他对设计方面的理解。当济华找我帮他的新书作推荐时，我感到非常高兴，同时也非常荣幸！《漫谈设计模式》绝对是一本看了之后让你对设计模式有一个全新理解的书。衷心希望这本书能为大家在设计模式上带来质的飞跃！

阅书如阅人

记得第一次见刘济华是在新加坡出差，当时一个瘦瘦的、戴着眼镜的小伙儿在我面前，起初我担心他是否能够快速上手。慢慢接触中，发现他虽然年龄小（我常称他为小孩），但能力不小：技术功底非常扎实、逻辑思维相当严谨，对一些技术问题有着独特的理解。通过阅读这本书，发现济华不光技术了得，语言组织功力也相当老练，让我对济华有了更深一步的了解。希望大家能够通过这本书了解他，读懂他，同时希望大家可以通过此书学有所成！

藏红岩
IBM 大中华区创新中心技术顾问

是程序员，也是作家

我认识 Co0der 已经两年了，但是我从来没想到他的另外一面——一位作家。我认为他的书籍不仅适合专家级别的 IT 读者，也适合入门级的读者以及介于这二者之间的 IT 读者，它有助于你对架构和设计模式有个全面的认识。我想推荐这本书籍给所有读者，希望你们能够阅读、理解以及实践其中所提及的方法，并在你的项目中获得成功。我希望 Co0der 一路顺风，再接再厉。

Kiran Kumar Pabbathi
Technology Consultant III in HP Bangalore

精妙的设计模式

济华是一个谦虚、沉稳、工作中追求完美并对技术有着执着追求的人。本书字里行间表现出严谨、一丝不苟的风格，结合作者自身工作中的经历和感悟，阐述了设计模式的概念和应用场景，希望读者朋友能从中体会到作者对设计模式选择和应用的精妙之处。

秦峰
百度软件工程师

编程习惯与编程态度

和 Co0der 共事过两年，某一天他 MSN 上给我留言说他出了一本 Design Pattern 的书，我抱着好奇心下载下来，花了一天时间一口气读完，我才改变了设计模式图书千篇一律的印象。这本书除了阐述一些基本的 OOP 编程原则，我觉得最重要的是推荐了一种编程习惯和编程态度，配合精简而优雅的实例代码，我想必定会为那些陷入设计模式泥潭的初学者带来一些灵感。

陆健
盛大在线基础研发部项目经理

选对正确的学习方向

刘济华做事喜欢把问题了解透彻，没有一点含糊的地方，正是因为这种严谨的做事态度，才有了本书的诞生。这是一本非常实用的书籍，希望这本书能给广大程序爱好者提供一个理解设计模式的正确渠道，使大家在编程中能够熟练运用设计模式思想，拓宽解决思路。

吕游
金蝶软件高级顾问

第一篇
模式介绍

漫谈设计模式——从面向对象开始

模式被引入软件开发和 OOP 语言的流行是分不开的，在 20 世纪 80 年代末至 90 年代初，面向对象软件设计逐渐成熟，被计算机界广泛理解和接受。然而由于专业开发人员和非专业开发人员作出的设计差异巨大，为了让 OOP 开发人员能够使用 OOP 设计进行专业开发，经过多年的不懈努力，最终由 GoF 四人根据当时的一些成熟经验和解决方案归纳出了 23 条最基本的设计模式，以供 OOP 开发人员学习和使用。

本篇首先从面向对象谈起，回顾一下面向对象的一些基础概念，讲述 OOP 开发设计带来的好处，以及为什么要学习和使用设计模式。接着在第 2 章将为读者讲述第一个最简单也最容易理解的模式——模板方法（Template Method）模式。

第 1 章　谈面向对象和模式

本章将简单总结一下面向对象领域的概念，读完本章并不能使你成为面向对象的专家，它将为你学习以下各章做准备。

主要包括内容如下：

- 面向对象的概念和范式。
- 模式的概念及与面向对象之间的关系。
- 学习设计模式常见的问题。

当你阅读完全书后，再回来阅读本章，你将会有更多体会，也一定能回答这些设计模式的常见问题了。