HZ BOOKS
华章IT

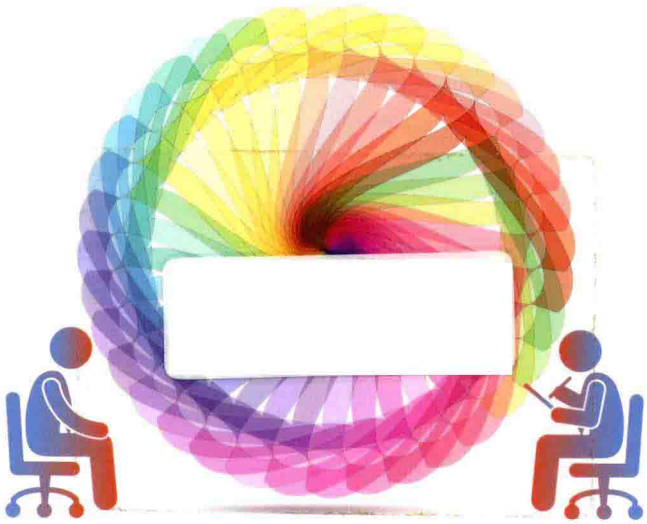
资深软件架构师撰写，深入浅出地阐释设计模式的概念、应用场景、方法及最佳实践

以GoF的23种设计模式为基础，详细解读如何正确选择和应用设计模式，涵盖常见设计模式相关面试问题

设计模式精解及面试攻略

[印度] 纳拉西姆哈·卡鲁曼希（Narasimha Karumanchi）
斯克林瓦萨·拉奥·梅达（Sreenivasa Rao Meda） 著

刘品杰 译

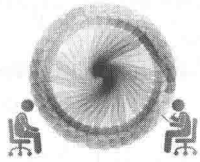


PEELING DESIGN PATTERNS

FOR BEGINNERS & INTERVIEWS

机械工业出版社
China Machine Press

设计模式精解及面试攻略



PEELING DESIGN PATTERNS

FOR BEGINNERS & INTERVIEWS

[印度] 纳拉西姆哈·卡鲁曼希（Narasimha Karumanchi）
斯克林瓦萨·拉奥·梅达（Sreenivasa Rao Meda） 著

刘品杰 译

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

设计模式精解及面试攻略 /（印度）卡鲁曼希（Karumanchi, N.），（印度）梅达（Meda, S. R.）著；刘品杰译 . —北京：机械工业出版社，2016.4
书名原文：Peeling Design Patterns: For Beginners & Interviews

ISBN 978-7-111-53615-4

I. 设… II. ① 卡… ② 梅… ③ 刘… III. 软件设计 IV. TP311.5

中国版本图书馆 CIP 数据核字（2016）第 085036 号

本书版权登记号：图字：01-2016-0686

设计模式精解及面试攻略

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码：100037）
责任编辑：秦　健　　责任校对：董纪丽
印　　刷：三河市宏图印务有限公司　　版　　次：2016 年 5 月第 1 版第 1 次印刷
开　　本：186mm × 240mm　1/16　　印　　张：16.25
书　　号：ISBN 978-7-111-53615-4　　定　　价：59.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88379426　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzit@hzbook.com

The Translator's Words 译者序

对于软件开发的新手来说，设计模式是什么？设计模式能帮助我做什么？也许这些问题你还无法回答得很清楚；也许你还在为 Lambda 表达式和反射等概念所困扰；也许你还在为能够弄清楚某种软件框架而苦恼，却正在你想要找一份工作的时候，被面试官问到设计模式的问题；也许你的软件开发团队在需求三番五次变更之后仍然要求按照时间完成软件开发，否则其他部门的同事就无法继续工作。没有办法！因为软件开发已经从三五个人的作坊式生产转变成了几万人的大规模流水线式的开发。虽然认证机构在讲述软件项目生命周期管理和开发流程时总把流程分得很明晰，但是实际开发的时候，却像是一场浑水摸鱼似的运动战。

在来自于 IBM Rational 软件的工程师们开始编写第一本关于设计模式的书籍时，他们所关心的就是如何提高大规模开发过程中设计人员整体的软件设计能力和软件编写质量。虽然当时的 IBM 还是一家以卖服务器为生的公司，但旗下的 Rational 品牌还是提供了一系列用于提高软件开发质量的软件，如版本管理工具 Clear Case 软件、缺陷管理与变更追踪软件 ClearQuest 等。这些软件很多已经成为大规模商业软件开发中必不可少的工具。当然，我在这里无意于为 IBM 做广告，我想说明的只是设计模式也是一种提高设计人员开发能力和软件编写质量的工具。它能够帮助你从与项目经理、QC、QA 无尽的沟通与争吵中解放出来，让你和其他开发人员和睦相处以更快地完成开发工作，协助你在需求漫天飞舞的环境中捋清头绪，让你有更多时间做一些对自己身心有益的事情。当有一天你真正地经历了这一次蜕变，成为软件开发专家的时候，望尽沧海，还是要感谢设计模式给我们带来的这一切。

本书的立意不仅仅在于介绍基本的设计模式知识，更多的是说明在何种条件下选择正确的设计模式。因为在实践中，滥用或错用设计模式有时候是更大的浪费。因此，作者在设计模式的介绍过程中穿插了不同类型模式之间的比较，并在第 8 章和第 9 章给出了大量

的 Java 问题实例，供读者深入理解。第 8 章的面试问题中给出的代码经译者测试，有些仍需要做一定的工作才能够满足运行条件。但第 8 章和第 9 章的示例与问题对于初学者明确 Java 中很多基本概念和代码实现是非常有益的，也能避免自己在日后的工作中陷入简单的错误中。

由于翻译时间有限，再加上译者的水平有限，难免存在不足或错误，还望各位读者不吝指正。

Preface 前　言

亲爱的读者，请先别着急往后翻！我知道你们很多人从来都不读前言。但是我强烈推荐大家在继续阅读之前先好好看一下前言。这是因为本书的前言会提供一些阅读本书所必需的知识。

首先，编写本书时我们假定你有一定量的计算机知识。其次，编写本书的主要目的并不是为大家提供一本设计模式的查询手册或技术面试指南。我们在编写本书之前就设立了以下目标：

- 本书的语言保证没有任何计算机软件编程背景的读者能够轻松和透彻地理解本书所表述的内容。
- 本书以简单直接的方法清晰地展示设计模式的核心思想。
- 在读完本书之后，每位读者都会比以往更加希望优化自己的软件设计，并且会更加乐于参与日常工作中的软件架构设计讨论。
- 为了让大家更好地理解设计模式，本书提供了足够多的代码示例。这些示例对软件工程师面试也非常有帮助。所以建议大家好好读一读设计面试问题的相关章节。
- 在读透本书之后，一般软件工程师应该都能达到软件架构师的水准。

在 20 世纪 80 年代末，由于软件设计的结果总是令人很不满意，设计模式才真正地被引入程序设计实践之中。在后续的发展过程中，很多抽象方法（算法和数据结构）对于流程化 / 函数式编程契合得非常完美。但是它们与面向对象编程的联系不多。

随后，一本介绍设计模式的书籍标志着软件设计历史的转折点。1995 年，4 位精通面向对象的软件设计师（Gamma、Helm、Johnson、Vlissides）出版了一本介绍 23 种设计模式的书籍[⊖]。这本获得了巨大成功的书，又被称为 GoF 手册。

设计模式不断地帮助初学者（新手）避免常见的错误，并激励高级程序员构建出更好的

⊖ 书名为《设计模式：可复用面向对象软件的基础》，书号为 978-7-111-07575-2，已由机械工业出版社引进出版。——编辑注

软件。

本书的编写更像为 GoF 提出的这些设计模式所举办的庆功会。最近几年，几乎所有面向对象的软件架构都构建在这些设计模式的基础上。

在本书中，为了便于学生和教师理解，我们使用简单例子来诠释设计模式。本书结尾给出了常见的软件设计面试问题，以帮助求职者提升面试表现。

跟随本书深入了解设计模式、软件过程和方法会帮助你开发出更好的应用软件和基本架构。通过阅读本书你能够全面了解这些关键的设计模式。我们自己从中也获益良多，相信各位读者也一定会如此。

如果每一位求职者都能够完成本书的学习并且充分理解本书的内容，那么我相信你一定能够征服面试官。这也是本书的编写目的之一。

本书也适用于软件工程的本科生和研究生的学习与学术研究。本书中的各个章节都包含了设计理论和相关问题。作为一名学生，你也能够通过阅读本书的内容以准备相关考试。本书对各个关键点都有详尽的介绍。

建议各位读者完整阅读本书至少一次，以便对各个知识点有一个大概理解。在后续的学习和阅读过程中，你就能够有的放矢。尽管如此，考虑到人为原因，本书肯定会有部分不实之处，通过不断阅读也能够鉴别书中的谬误。如果你发现了任何错误，我们希望你能够到 www.CareerMonk.com 上传相关内容。也建议你在阅读本书的过程中多访问该网站，以查看新发现的问题或者任何更正。我们热忱地欢迎你对本书提出宝贵意见：Info@CareerMonk.com。

祝好，我们相信你会发现本书物有所值。

Sreenivasa Rao Meda

教授，博士

Narasimha Karumanchi

CareerMonk.com 创始人

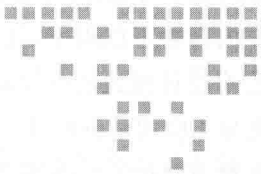
Acknowledgements 致　谢

首先，感谢我们的家人和亲爱的朋友们。他们对我们生活的支持和鼓励让我们有能力完成本书的编写。

我们还要通过这本书对那些审读本书的朋友表示感激之情，包括对所有为本书提供各种支持，探讨技术问题，提供阅读、编写和修改意见，允许本书引用相关简介，并协助编辑校对和设计的朋友。我们尤其需要感谢下列人士：

- Kalyani Tummala，IIT 克勒格布尔分校，Xilinx 公司
- Girish P. Saraph 教授，Vegayan 系统公司创始人
- Manoj Patra，微软印度公司高级经理
- A. Vamshi Krishna，IIT 坎普尔分校，Mentor 图形公司
- Rambabu Dubbukuri，IIT 坎普尔分校，微软印度公司
- Venkata Ramana Sanaka，诺基亚公司
- Kishore Jinka，IIT 孟买分校
- Vikas Kedia，IIT 孟买分校，谷歌印度公司
- Suman Somavarapu，IIT 孟买分校，De-Shaw 印度公司
- Anil Bhat，IIT 罗克分校，微软印度公司
- Chaganti Siva Rama Krishna Prasad，StockMonks 公司创始人
- Kumar 和 Jagan，Impression Design Studio 公司创始人
-

目　录 *Contents*



第 1 章 Chapter 1

概　述

1.1 全书概览

设计模式是一种针对软件设计问题的形式化解决方案。它能够在不同场景下应用，并成功地解决在特定条件下会反复发生的设计问题。需要强调的是，设计模式并**不是**一种**发明创造**。

设计模式是在众多软件工程实践过程中不断观察和学习而来的，它是能够解决问题的形式化的最佳实践。

设计模式以专业软件开发人员的经验为总结形成了一系列系统性的解决方案。这其中包含了最常见的设计问题、解决方法以及解决方法对软件开发的影响。

本书包含如下内容：

- 为什么设计模式对于面向对象的软件开发和设计如此重要？
- 设计模式的基本形式和分类有哪些？
- 应该在什么时候使用设计模式？
- 如何应用设计模式？
- 常见的设计模式相关的面试问题。

设计模式提供了一种解决问题的基本框架。在我们解决实际问题的过程中，我们必须考虑可能存在着多种解决方法，以及其中是否有适配特定设计模式的可能。

理解并实际应用设计模式其实是一件非常困难的事。因为它并不仅仅需要我们学习设

计模式，更重要的是学习设计模式的实际应用方法。

本书的主要内容并不仅限于对设计模式的基本介绍。与其他介绍设计模式书籍所不同的是，本书不是对设计模式的一个简单的分类总结，而是介绍了一种对软件设计问题的分解方法并使之更易于应用某种模式。

同时本书也对面向对象的程序设计方法、UML 建模语言和设计模式常见的面试问题进行了详细介绍。

1.2 设计模式简史

模式这个词最初是由土木工程师 Christopher Alexander 提出的一种建筑学概念。1987 年，Kent Beck 和 Ward Cunningham 首先将模式的概念应用在程序设计中，并在当年的一次会议中发表了相关成果。在之后的若干年中，Beck、Cunningham 等人又继续发展了相关的理论。

设计模式的概念在计算机科学领域的普及主要归功于 1994 年《Design Patterns: Elements of Reusable Object-Oriented Software》[⊖]一书的出版。这本书由软件设计领域的 4 位世界顶级大师（Erich Gamma、Richard Helm、Ralph Johnson、John Vlissides，又称 Gang of Four 或 GoF）所合著。

1.3 本书读者对象

这本书主要写给那些想要通过学习设计模式来提高面向对象编程开发技巧的程序员，以及想通过这些技巧进行求职的程序员。

在完成本书各个章节的学习之后，你应该能够：

- ❑ 理解什么是设计模式，并能够对多种设计模式进行描述和分类。
- ❑ 能够使用设计模式中的各种专业词汇，并用于理解和讨论面向对象的程序设计。
- ❑ 理解最常见的几类设计模式，并能够进行应用。
- ❑ 理解在技术面试过程中最常见的设计模式面试问题。

在阅读本书之前，你需要熟悉 Java 语言和面向对象的基本概念，如多态、继承、封装等。

如果你熟悉统一建模语言（UML）的基本知识，那将会对学习本书有一定帮助，但这不是学习本书的必要条件。本书将会介绍 UML 的基本知识。

⊖ 本书中文版《设计模式：可复用面向对象软件的基础》已由机械工业出版社引进出版，ISBN：978-7-111-07575-7。

1.4 本书对面试的帮助

本书是一本旨在帮助软件工程师应聘软件开发职位和帮助面试官筛选应聘者的书。同时它还包含了基本概念和解决的设计问题。

它涵盖了必需的基本概念，如 UML 符号，并详细介绍了所有的基本设计模式。

它也给出了帮助读者理解的提示和参考。本书的作者团队具有广泛的学术和行业经验。他们已经发表了许多关于算法的论文，在谷歌、微软、亚马逊以及一些软件创业公司中积累了实践经验，并参与了众多软件开发职位的面试工作。

1.5 如何阅读本书

建议对本书至少通读两遍。在第一遍阅读时，你会在常见的软件架构中注意到并识别这些模式。

在第二遍阅读时，你会发现这些模式在自己的软件设计中所产生的帮助，并会发现本书不包括的新设计模式。

一旦你熟悉了模式的概念，就可以创造出自己所需要的模式。这会让你在以后的工作中获益良多。

本书最有价值的贡献之一是，它不仅能够让你识别特定的模式，而且能够让你学会在不同的情况下选择适当的模式。

这样，在以后的阅读中你就能够直接翻到特定的章节去查询你所需要的知识。

1.6 本书主要内容

编写本书的主要目的是使用简单且易于理解的举例方式来展示设计模式。本书讨论的所有设计模式都出自 GoF 的经典著作。

此外，本书还包括了软件设计各个方面的概念（设计面试问题、Java 面试问题、MVC 模式等）。

本书各章节安排如下：

第 2 章：学习后续章节所必需的 UML 基本介绍和必要概念。

第 3 章：对设计模式和模式的分类等概念进行介绍。

第 4 章：讨论创造型模式（抽象工厂、工厂方法、生成器、原型和单例等模式）。

第 5 章：讨论结构型模式（适配器、桥接、组合、装饰、门面、享元和代理等模式）。

第 6 章：讨论行为型模式（职责链、命令、解释器、迭代器、中介者、备忘录、观察

者、状态、策略、模板方法、访问者等模式)。

第 7 章：提供所有设计模式的汇总，并向初学者提供一些提示。

第 8 章：包括常见的面试问题及实例。

第 9 章：涵盖 Java 面试问题和其他的一些概念，如 MVC 模式等。

我们通过在 Java 语言中实现该设计模式的一个示例来引入对每种设计模式的讨论。然后通过对代码段和 UML 图的分析去学习给定的模式是如何在实例中应用的。

在学习完每个模式之后，我们会提出一些问题供读者解答，以便读者能够深化对该模式的理解。我们会在可以应用模式的任何地方对相似的几种模式进行对比。

本书中的例子会尽量保持简单以便读者理解。这样做的目的是让读者能更好地理解每个模式中的示例和解释。

第 2 章对统一建模语言（UML）进行了概述，并讨论了类和时序图中的各种元素。

第 8 章讨论在采用不同的设计模式时可能遇到的一些实际问题。这一章介绍了如何在设计中使用各种模式，以及常见的面试问题。

1.7 源代码免责声明

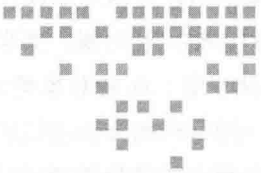
本书作者和出版商均不能承诺或担保本书中公布的或是暗含的软件的适用性，包括但不限于适销性的默示保证、针对特定用途的适用性或非侵权行为。

本书作者和出版商均不承担由于使用、修改或分发本书介绍的软件或其衍生物所造成的任何赔偿责任。

1.8 本书中使用的工具

本书中的例子都是使用 Java 语言编写的。读者应该可以且具备直接阅读代码的能力。但是如果你想要运行这些代码，则至少需要一个 Java 开发环境。

你需要一个简单的文本编辑器（如在 UNIX 环境中的 vi 或者 Windows 环境中的记事本）和 Java SDK（版本 1.2 或更高版本）。除此之外，你还需要一些能够创建 UML 图的工具（例如 Start UML）。



第 2 章 Chapter 2

UML 基础

2.1 什么是 UML

UML 是统一建模语言（Unified Modeling Language）的缩写。它最初用于描述复杂的软件和非软件系统的行为。现今 UML 已经成为一个通行标准。UML 伴随着面向对象的概念和方法而生。以面向对象思想设计的软件系统也通常使用图像语言进行建模。

UML 图也能够从不同的角度（如设计、实施、部署等）展示建模结果。这是因为 UML 是一种非常便于阐述系统的架构、行为和基本结构的建模语言。

作为标准的建模语言，UML 并不是一个软件开发过程。UML 也不同于其他常见的编程语言（如 C++、Java 和 COBOL 等）。UML 本身并不是一种编程语言，但通过构建 UML 图能够帮助我们很轻松地使用各种语言编写实际代码。

作为一种可视化的建模语言，UML 能够描述软件系统的需求说明和系统架构，并做到可视化和文档化。虽然 UML 常用于软件系统建模，但它也常用于非软件系统建模，如说明生产机构的工艺流程等。

2.2 为什么使用 UML

对象是面向对象世界的核心。面向对象的软件分析和设计中，最基本的需求是高效地识别对象。完成对象识别之后，赋予每个对象相应的功能或职责。在完成以上基本分析工

作之后，只需要根据分析输入完成代码设计即可。

话说无图无真相，有时候一张图片胜过千言万语。这句话非常适合形容 UML 的特点。在计算机软件设计中，面向对象概念的出现远比 UML 要早得多。所以在很长的一段时间内，一直缺少一种标准的方法来组织和固化面向对象的设计。就在这时，UML 走进了大家的视野，并且在面向对象分析和设计中起到了非常关键的作用。从此，UML 图就一直用于软件模型的构建。可以说，UML 发挥了不可替代的作用。

2.3 UML 符号

UML 符号是建模过程中最重要的元素。合理有效地使用 UML 符号对于实现一个完整可用的模型来说是至关重要的。如果你对模型的描述是错误的，那么对于软件设计来说建模结果是有百害而无一利的。

所以在开始学习本书之前，应该先学会如何使用 UML 符号。不同的符号代表了不同的事物和关系。UML 图也是由这些符号所代表的物件和关系组成的。

2.4 面向对象概念

由于 UML 在面向对象的分析和设计中长期扮演着重要的角色，所以首先让我们回顾一下面向对象的概念。对象通常包含了数据和操作数据的方法（又称为函数）。数据表示对象的状态。这样，我们使用一个**类**来描述特定的对象，并形成层次结构来模拟真实世界的系统。

这种层次结构表示为继承，或者说是类根据不同的需要扩展出不同的行为方式。

关于 Java 的基本概念和常见面试问题请参考第 9 章。

现实世界中存在于我们周围的实体对象和面向对象中抽象、封装、继承、多态的基本概念都可以使用 UML 进行表示。

因此，UML 强大到足以表征所有存在于面向对象分析和设计中的概念。UML 图就是其中专门用于表示面向对象中各种概念的工具。所以在学习 UML 之前，大家最好能掌握面向对象中的各种概念。以下就是面向对象思想的一些核心概念：

- **对象**：对象是理解面向对象思想的关键。现在让我们环顾四周，目所及处正是真实世界中各个对象的实例：狗、桌子、电视、自行车等。

现实世界中的对象有两个共通的特点：它们都具有状态和行为（功能）。狗有状态（名