Broadview
www.broadview.com.cn

高等院校软件专业方向 系列教材

设计模式

（Java版）

青岛东合信息技术有限公司 编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

高等院校软件专业方向系列教材

内 容 简 介

本书从最基本的设计原理及思想出发，深入讲解和剖析了23种常见的设计模式，每种模式都对应相应的案例，这些案例通俗易懂、围绕模式的核心思想，便于读者进一步理解和学习设计模式。全书共有8章，分别介绍了设计模式概述、设计原则、创建型模式、结构型模式、行为型模式、混合模式以及设计模式之间的对比。书中涉及了6大设计原则、23种设计模式，每种设计模式都从定义、应用以及实例这三个方面进行详细介绍。

本书重点突出、偏重应用，结合理论篇的实例讲解、剖析及实现，使读者能迅速理解并掌握知识，全面提高动手能力。

适应面广，可作为本科计算机科学与技术、软件外包专业、高职高专计算机软件、计算机网络、计算机信息管理、电子商务和经济管理等专业的程序设计课程的教材。

图书在版编目（CIP）数据

设计模式：Java版 / 青岛东合信息技术有限公司编著. —北京：电子工业出版社，2012.1
高等院校软件专业方向系列教材
ISBN 978-7-121-15582-6

Ⅰ. ①设… Ⅱ. ①青… Ⅲ. ①JAVA语言－程序设计－高等学校－教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字（2011）第271262号

策划编辑：张月萍
责任编辑：徐津平
文字编辑：张丹阳
印　　刷：北京市顺义兴华印刷厂
装　　订：三河市双峰印刷装订有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　　邮编：100036
开　　本：787×1092　1/16　　印张：16.75　　字数：375千字
印　　次：2012年1月第1次印刷
定　　价：39.00元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

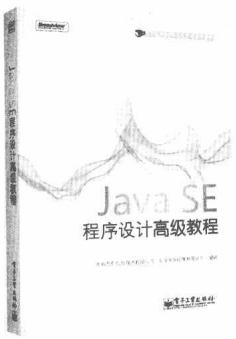
质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件到dbqq@phei.com.cn。

服务热线：（010）88258888。

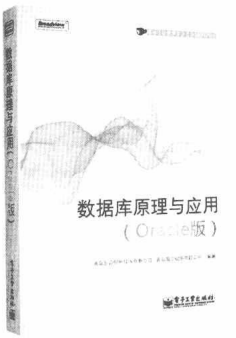
高等院校软件专业方向 系列教材

ISBN：978-7-121-12513-3
定价：49.00元

ISBN：978-7-121-11274-4
定价：46.00元



ISBN：978-7-121-11268-3
定价：46.00元

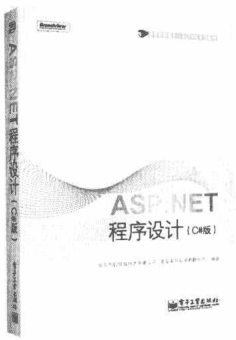


ISBN：978-7-121-11269-0
定价：59.00元

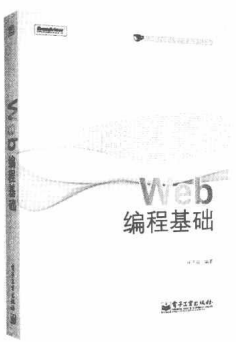
ISBN：978-7-121-12518-8
定价：49.00元

ISBN：978-7-121-13554-5
定价：59.00元

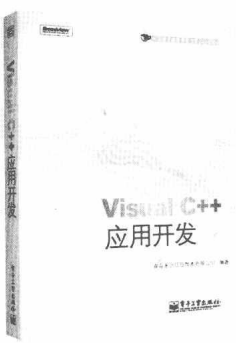
ISBN：978-7-121-13545-3
定价：39.00元



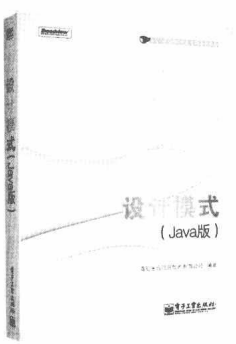
ISBN：978-7-121-13470-8
定价：69.00元



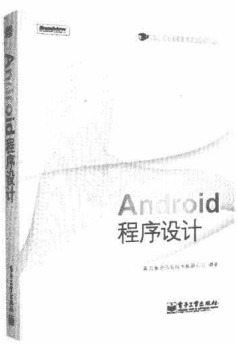
ISBN：978-7-121-12501-0
定价：43.00元



ISBN：978-7-121-15570-3
定价：69.00元



ISBN：978-7-121-15582-6
定价：39.00元



ISBN：978-7-121-15496-6
定价：49.00元

编　委　会

前　言

随着IT产业的迅猛发展，企业对应用型人才的需求越来越大。“全面贴近企业需求，无缝打造专业实用人才”是目前高校计算机专业教育的革新方向。

该系列教材是面向高等院校软件专业方向的标准化教材。教材研发充分结合软件企业的用人需求，经过了充分的调研和论证，并充分参照多所高校一线专家的意见，具有系统性、实用性等特点。旨在使读者在系统掌握软件开发知识的同时，着重培养其综合应用能力和解决问题的能力。

该系列教材具有如下几个特色。

1. 以应用型人才为导向来培养学生

强调实践：本系列教材以应用型软件及外包人才为培养目标，在原有体制教育的基础上对课程进行了改革，强化“应用型”技术的学习，使学生在经过系统、完整的学习后能够达到如下要求：

- 具备软件开发工作所需的理论知识和操作技能，能熟练进行编码工作，并掌握软件开发过程的规范；
- 具备一定的项目经验，包括代码的调试、文档编写、软件测试等内容；
- 相当于一年的软件开发经验。

2. 以实用技能为核心来组织教学

二八原则：遵循企业生产过程中的“二八原则”，即企业生产过程中80%的时间在使用20%的核心技术，强调核心教学，即学生在学校用80%的学习时间来掌握企业中所用到的核心技术，从而保证对企业常用技术的掌握。教材内容精而专，同时配以知识拓展和拓展练习，以满足不同层次的教学和学习需求。

3. 以新颖的教材架构来引导学习

自成体系：本系列教材采用的教材架构打破了传统的以知识为标准编写教材的方法，采用“全真案例”和“任务驱动”的组织模式。

- **理论篇**：即最小教学集，包含了“二八原则”中提到的常用技术，以任务驱动引导知识点的学习，所选任务不但典型、实用，而且具有很强的趣味性和可操作性，引导学生循序渐进地理解和掌握这些知识和技能，培养学生的逻辑思维能力，掌握利用开发语言进行程序设计的必备知识和技巧。
- **实践篇**：多点于一线，以完整的具体案例贯穿始终，力求使学生在动手实践的过程中，加深课程内容的理解，培养学生独立思考和解决问题的能力，并配备相关知识的拓展讲解和拓展练习，拓宽学生的知识面。
- **结构灵活**：本系列教材在内容设置上借鉴了软件开发中“低耦合高内聚”的设计理念，组织架构上遵循软件开发中的 MVC 理念，即在课程的实施过程中各高校可根据自身的实际情况（课程配比、时间安排、学生水平、教学情况等），在保证最小教学集的前提下可对整个课程体系进行横向（章节内容）、纵向（章节）裁剪。

4. 提供全面的教辅产品来辅助教学实施

为充分体现“实境耦合”的教学模式，方便教学实施，另外还开发了可配套使用的项目实训教材和全套教辅产品，可供各院校选购。

项目篇：多线于一面，以辅助教材的形式，提供适应当前课程（及先行课程）的综合项目，遵循软件开发过程，进行讲解、分析、设计、指导，注重工作过程的系统性，培养学生解决实际问题的能力，是实施“实境”教学的关键环节。

立体配套：为适应教学模式和教学方法的改革，本系列教材提供完备的教辅产品，主要包括教学指导、实验指导、电子课件、习题集、实践案例等内容，并配以相应的网络教学资源。教学实施方面，提供全方位的解决方案（课程体系解决方案、实训解决方案、教师培训解决方案和就业指导解决方案等），以适应软件开发教学过程的特殊性。

本系列教材由青岛东合信息技术有限公司研制，历时两年，参与编著的有韩敬海、丁春强、赵克玲、高峰、张幼鹏、张旭平、孙更新、孙云彩、曹宝香、崔文善、王成端等。本书的特约策划人为吕蕾女士。参与本书编写工作的还有：青岛农业大学、潍坊学院、曲阜师范大学、济宁学院、济宁医学院等高校，期间得到了各合作院校专家及一线教师的大力支持和协作。在此技术丛书出版之际要特别感谢给予我们开发团队大力支持和帮助的领导及同事，感谢合作院校的师生给予我们的支持和鼓励，更要感谢开发团队每一位成员所付出的艰辛劳动。如有意见或建议，可访问我公司网站（http://www.dong-he.cn）或发邮件至 dh_iTeacher@126.com。

iTeacher@ 教研组

2011 年 12 月

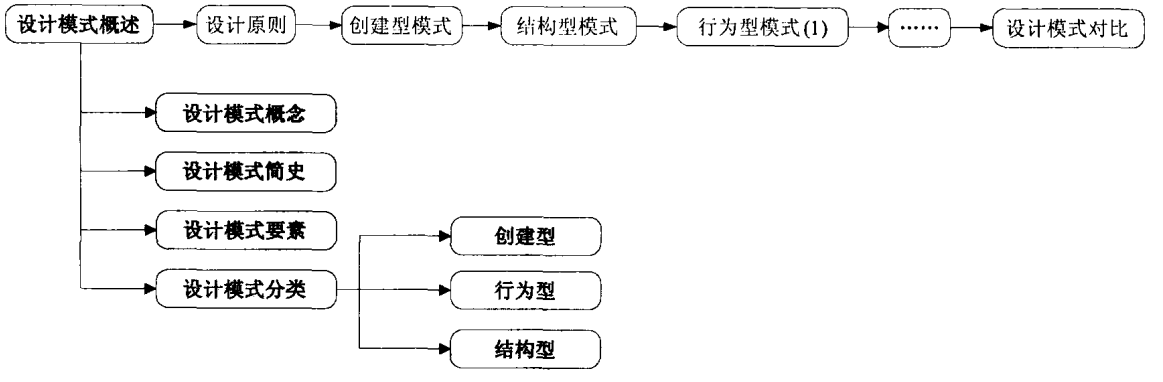
目　录

第 1 章　设计模式概述

本章目标

- 了解设计模式的概念
- 了解设计模式的历史
- 理解设计模式的要素
- 掌握设计模式的分类

学习导航



1.1 设计模式概念

设计模式（Design Pattern）是一套被反复使用、多数人知晓、经过分类编目的优秀代码设计经验的总结。使用设计模式是为了重用代码、使代码更易理解并保证代码的可靠性。毫无疑问，设计模式的使用于己于他人于系统都是有利的，设计模式使代码编制真正工程化，是软件工程的基石，使人们可以更加简单方便地复用成功的设计和体系结构，将已证实的技术表述成设计模式也会使新系统开发者更加容易理解其设计思路。在面向对象编程语言中（例如 Java），设计模式为我们提供了一套可复用的面向对象技术。

Java 提供了丰富的 API，使编程似乎变成了类似堆积木的简单“拼凑”和“调用”，甚至有人提倡“蓝领程序员”，这些都是对现代编程技术的不了解所致。可复用面向对象软件系统一般划分为两大类：应用程序工具箱和框架（Framework）。我们平时开发的具体软件都是应用程序，Java 的基础类库属于工具箱，而框架是构成特定软件可复用设计的一组相互协作的类。框架通常定义了应用体系的整体结构类和对象的关系等设计参数，以便具体应用实现者能集中精力于应用本身的特定细节。框架主要记录软件应用中共同的设计决策，强调设计复用，因此成熟的框架设计中必然要使用设计模式，如果熟悉这些设计模式将有助于对框架结构的理解，从而能够迅速掌握框架的结构。例如，初次接触 EJB、Java EE 等框架，会觉得特别难学，难掌握，此时如果先掌握设计模式，则使我们具有剖析 EJB 或 Java EE 系统的能力。

Java 设计模式贯彻的原理是：面向接口编程，而不是面向实现。其目标原则是：降低耦合，增强灵活性。

1.2 设计模式简史

设计模式的研究起源于建筑工程设计大师 Christopher Alexander 的关于城市规划和建筑设计的著作。尽管他的著作是针对城市规划和建筑设计的，但是其观点实际上适用于所有工程设计领域，包括软件开发设计领域。Alexander 在其著作中指出，使用现在的设计方法所设计出的建筑物，不能满足所有工程设计的基本目的：改善人类的条件。Alexander 想要发明的建筑结构，是能使人类在舒适和生活质量上受惠的建筑结构。他得出的结论是，设计师必须不断努力，以创造出更加适合所有住户、用户和他们的社区的结构，从而满足他们的需要。同样软件开发中的设计模式也是不断进行研究、创新，以更加适合软件工程的各个方面。

设计模式在软件行业中的应用可以追溯到 1987 年。Ward Cunningham 和 Kent Beck 在一起用 Smalltalk 做设计用户界面的工作，他们决定使用 Alexander 的理论发展出一个有五个模式的语言来指导 Smalltalk 的新手，因此他们写了一篇“Using Pattern Languages for Object-Oriented Programs（使用模式语言做面向对象的程序）”的论文。此后不久，James O. Coplien 开始搜集 C++语言的成例（成例也可认为是一种设计模式，更偏重于编码技巧），这些 C++成例发表在 1991 年出版的《Advanced C++ Programming Styles and Idioms（高级 C++

编程风格和成例)》一书中。

1990 年到 1992 年，Erich Gamma、Richard Helm、Ralph Johnson 和 John Vlissides 四个人（常被称为 Gang of Four、GoF 或“四人帮”，如图 1-1 所示），开始了搜集模式的工作。1993 年 8 月，Kent Beck 和 Grady Booch 主持了在美国科罗拉多的山区度假村召开的第一次关于模式的会议，模式研究的主要人物都参加了这次会议，包括 James O. Coplien、Doug Lea、Desmond D'Souze、Norm Kerth、Wolfgang Pree 等。

1995 年 GoF 的“*Design Patterns: Elements of Reusable Object-Oriented Software*”（《设计模式：可复用面向对象软件的基础》）出版。该书第一次将设计模式提升到理论高度，并将之规范化，同时提出了 23 种基本设计模式。此书发表之后，带动了设计模式的研究热潮，被确定为模式结构的数目也呈爆炸性增长。时至今日，在可复用面向对象软件的发展过程中，新的设计模式仍然不断出现。

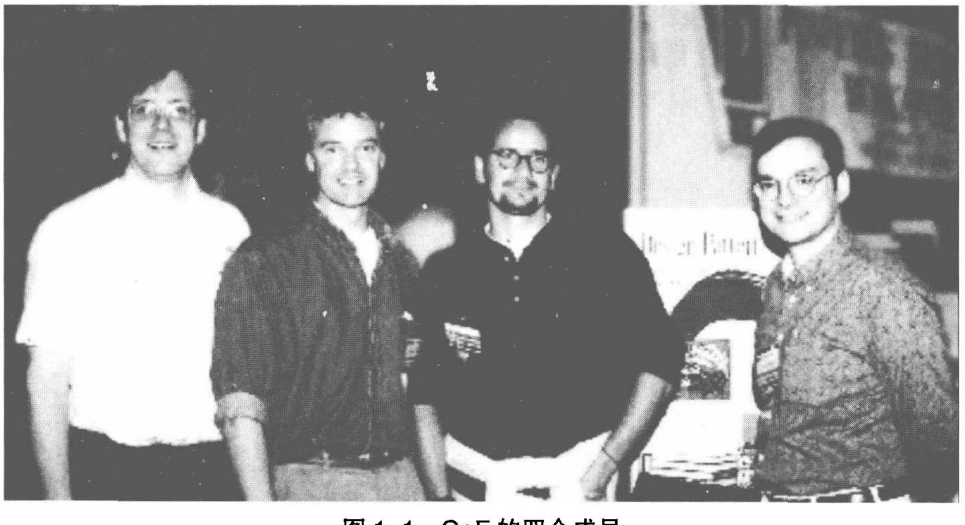


图 1-1　GoF 的四个成员

1.3　设计模式要素

描述设计模式需要一定的格式，可采用的格式有很多，在 Alexander 的著作中使用的格式就叫做 Alexander 格式，在 GoF 的著作中使用的格式就叫做 GoF 格式。在 GoF 格式中，沿用了一些 Alexander 格式的段落标题，这些标题常常叫做正则格式。尽管在不同的格式中，正则格式的细节有所不同，但设计模式应当包含以下几个要素。

- 模式名称（pattern name）

  设计模式的名称简洁地描述了设计模式的问题、解决方案和效果。一个模式必须有一个有意义的、简短而准确的名字。好的模式名称便于设计人员之间交流思想，进行抽象讨论及研究设计结果。找到恰当的模式名称也是设计模式编目工作的难点之一，命名一个新的模式，就可以讨论模式并在编写文档时使用它们。

- 问题（problem）

  描述了应该在何时使用模式。它解释了设计问题和问题存在的前因后果，它可能描述了特定的设计问题，如怎样用对象表示算法等，也可能描述了导致不灵活设计的类或对象结构。有时候，问题部分会包括使用模式必须满足的一系列先决条件。

- 环境或初始环境（context 或 initial context）

  环境说明模式的使用范围，也是模式应用之前的起始条件（也叫前提条件）。

- 解决方案（solution）

  描述了设计的组成成分，它们之间的相互关系及各自的职责和协作方式。模式就像一个模板，可应用于多种不同场合，因此解决方案并不描述一个特定而具体的设计或实现，而是提供设计问题的抽象描述和怎样用一个具有一般意义的元素组合（类或对象组合）来解决这个问题。

- 效果（consequences）

  描述了模式应用的效果及使用模式应权衡的问题。效果用来描述设计模式的利弊，它往往是衡量模式是否可用的重要因素，对于评价设计选择和理解使用模式的代价及好处具有重要意义。软件效果大多关注对时间和空间的衡量，表述了语言和实现问题。因为复用是面向对象设计的要素之一，所以模式效果包括其对系统的灵活性、扩展性或可移植性的影响，显式地列出这些效果对理解和评价这些模式具有很大的帮助。

- 举例（examples）

  使用一个或多个示意性的应用来说明特定的真实环境，以及模式是如何应用到环境中、改变环境并且给出当模式结束时的末态环境。例子有助于理解模式的使用方法和适用性，每一个例子均可以附带一个实现的样本，说明解答是如何给出来的。从熟知系统里取出来的、有视觉效果的，或以比喻方式表达的例子，更易于使用者理解。

- 末态环境（resulting context）

  模式应用到系统之后的状态。末态环境包括模式带来的好结果和坏结果，以及新状态中含有的其他问题和可能涉及的其他有关系的模式。末态环境是模式的末态条件和可能有的副作用。描述末态环境可以帮助比较末态环境与起始环境的区别和联系。

- 推理（rationale）

  推理解释本模式的步骤、规则，以及此模式作为一个整体是如何以特定的方式解决模式的。推理让使用者知道模式是如何工作的，为什么可以工作，以及使用此模式的优点是什么。模式的解答描述模式外部的、可见的结构和行为，而推理则给出模式在系统表层以下的深层结构和关键机制。

- 其他有关模式（related pattern）

  描述在现有的系统中此模式与其他模式的静态和动态的关系。相关模式的初始环境和末态环境经常是相容的，这些模式有可能是本模式的前任模式，即应用了这些模