

21世纪高等学校规划教材 | 电子商务



Java设计模式

于卫红 编著



清华大学出版社

21世纪高等学校规划教材 | 电子商务



Java设计模式

于卫红 编著

清华大学出版社
北京

内 容 简 介

本书围绕面向对象的三大机制(封装、继承、多态)和面向对象的基本原则,从介绍设计模式的基本概念出发,结合作者多年教学与项目开发的实际经验,从 GoF 的 23 种常见设计模式中精心挑选了 17 种进行详细讲解。每种模式都对应了电子商务系统开发中的应用案例。这些案例易于理解,紧紧围绕模式的核心思想,具有实用性和启发性,有助于读者不断领悟设计模式的内涵。

全书分为 4 个部分,共 23 章,内容涉及了设计模式基础、创建型设计模式(包括工厂方法模式、抽象工厂模式等 5 种)、结构型设计模式(装饰者模式、适配器模式等 6 种)、行为型设计模式(包括观察者模式、策略模式等 6 种),每种模式都从动机、定义、结构、实例、应用效果这几方面进行详细介绍,每章都配有大量的习题,附录部分给出了习题答案。

本书层次结构清晰,通俗易懂、实用性强,最大特色是通过介绍各种模式在电子商务系统开发中的应用案例,起到了抛砖引玉的作用,启发读者在“互联网+”的时代背景下进一步思考设计模式在 Web 应用、电子商务、办公自动化、商务智能等系统开发中的灵活运用。

本书适用面广,可作为本科或高职高专电子商务、计算机科学与技术、信息系统与信息管理等专业的程序设计类课程的教材,也是设计模式自学者的理想用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Java 设计模式/于卫红编著.--北京:清华大学出版社,2016

21 世纪高等学校规划教材.电子商务

ISBN 978-7-302-43781-9

I. ①J… II. ①于… 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(2016)第 100107 号

责任编辑:闫红梅 李 晔

封面设计:傅瑞学

责任校对:时翠兰

责任印制:宋 林

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>,010-62795954

印 刷 者:北京市人民文学印刷厂

装 订 者:三河市溧源装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:23.75 字 数:574 千字

版 次:2016 年 10 月第 1 版 印 次:2016 年 10 月第 1 次印刷

印 数:1~2000

定 价:49.00 元

目 录

第一部分 设计模式基础

第 1 章 设计模式概述	3
1.1 设计模式的定义与起源	3
1.1.1 模式与设计模式	3
1.1.2 模式的起源	3
1.1.3 设计模式的基本要素	4
1.2 为什么要使用设计模式	6
1.2.1 设计模式的作用	6
1.2.2 如何正确地学习和使用设计模式	7
1.3 GoF 的 23 种设计模式	8
1.3.1 设计模式的分类	8
1.3.2 23 种设计模式的简单描述	9
1.4 设计模式与架构、框架的区别与联系	10
1.4.1 什么是架构	10
1.4.2 什么是框架	12
1.4.3 设计模式、架构、框架的比较	12
1.5 本章习题	13
第 2 章 设计模式与 UML 建模	15
2.1 UML 建模简介	15
2.2 UML 类图	16
2.2.1 类与类图	16
2.2.2 类图中的基本元素符号	16
2.2.3 类与类之间的关系	18
2.3 UML 建模实例分析	22
2.3.1 实例分析 1——聊天系统登录模块	23
2.3.2 实例分析 2——某 C/S 软件注册模块	24
2.4 本章习题	26
第 3 章 面向对象的设计原则	29
3.1 软件设计的“七宗罪”	29

3.2	面向对象的三大机制	31
3.2.1	封装	31
3.2.2	继承	33
3.2.3	多态	34
3.3	面向对象基本原则	35
3.3.1	单一职责原则	35
3.3.2	开闭原则	36
3.3.3	里氏替换原则	38
3.3.4	接口隔离原则	41
3.3.5	依赖倒置原则	43
3.4	本章习题	45

第二部分 创建型设计模式

第4章	工厂方法模式	49
4.1	从简单工厂模式到工厂方法模式	49
4.1.1	简单工厂模式	49
4.1.2	工厂方法模式	52
4.2	工厂方法模式的结构及其代码示例	53
4.2.1	工厂方法模式的结构	53
4.2.2	工厂方法模式的代码示例——笔记本电脑生产的模拟	54
4.3	工厂方法模式在电子商务系统中的应用	58
4.3.1	问题描述	58
4.3.2	使用工厂方法模式实现消息通知的发送	61
4.3.3	工厂方法模式的应用效果分析	67
4.4	本章习题	68
第5章	抽象工厂模式	71
5.1	抽象工厂模式概述	71
5.1.1	抽象工厂模式的动机	71
5.1.2	抽象工厂模式的定义	72
5.2	抽象工厂模式的结构及其代码示例	73
5.2.1	抽象工厂模式的结构	73
5.2.2	抽象工厂模式的代码示例——笔记本电脑生产商业务的拓展	74
5.3	抽象工厂模式在电子商务系统中的应用	78
5.3.1	问题描述	78
5.3.2	抽象工厂模式在电子商务系统数据访问层的使用	78
5.3.3	抽象工厂模式的应用效果分析	85
5.4	本章习题	86

第 6 章 建造者模式	88
6.1 建造者模式概述	88
6.1.1 建造者模式的动机	88
6.1.2 建造者模式的定义	88
6.2 建造者模式的结构及其代码示例	89
6.2.1 建造者模式的结构	89
6.2.2 建造者模式的代码示例——计算机生产过程的模拟	90
6.3 建造者模式在电子商务系统中的应用	93
6.3.1 问题描述	93
6.3.2 使用建造者模式实现电子商务系统中的数据导出	94
6.3.3 建造者模式的应用效果分析	100
6.4 本章习题	101
第 7 章 单例模式	103
7.1 单例模式概述	103
7.1.1 单例模式的动机	103
7.1.2 单例模式的定义	104
7.2 单例模式的结构及其代码示例	104
7.2.1 单例模式的结构	104
7.2.2 单例模式的代码示例——Windows 回收站的创建模拟	104
7.3 单例模式在电子商务系统开发中的应用	110
7.3.1 问题描述	110
7.3.2 使用单例模式编写网站计数器程序	110
7.3.3 单例模式的应用效果分析	112
7.4 本章习题	112
第 8 章 原型模式	115
8.1 原型模式概述	115
8.1.1 原型模式的动机	115
8.1.2 原型模式的定义	116
8.2 原型模式的结构及其代码示例	116
8.2.1 原型模式的结构	116
8.2.2 原型模式的代码示例——简历克隆	117
8.3 原型模式在电子商务系统中的应用	121
8.3.1 问题描述	121
8.3.2 使用原型模式实现广告邮件的克隆	121
8.3.3 原型模式的应用效果分析	125
8.4 本章习题	126

第9章 创建型模式总结	128
-------------	-----

9.1 创建型模式小结	128
9.2 创建型模式综合练习	130

第三部分 结构型设计模式

第10章 装饰者模式	141
------------	-----

10.1 装饰者模式概述	141
10.1.1 装饰者模式的动机	141
10.1.2 装饰者模式的定义	142
10.2 装饰者模式的结构及其代码示例	142
10.2.1 装饰者模式的结构	142
10.2.2 装饰者模式的代码示例——早餐店订单系统	143
10.3 装饰者模式在电子商务系统中的应用	148
10.3.1 问题描述	148
10.3.2 使用装饰者模式实现电子商务系统的计价功能	148
10.3.3 装饰者模式的应用效果分析	156
10.4 本章习题	156

第11章 适配器模式	159
------------	-----

11.1 适配器模式概述	159
11.1.1 适配器模式的动机	160
11.1.2 适配器模式的定义	160
11.2 适配器模式的结构及其代码示例	160
11.2.1 类适配器模式的结构	161
11.2.2 类适配器模式的代码示例——学生管理系统的维护	161
11.2.3 对象适配器模式的结构	164
11.2.4 对象适配器模式的代码示例——插座转换器的模拟	164
11.2.5 对象适配器模式和类适配器模式的比较	166
11.3 适配器模式在电子商务系统中的应用	167
11.3.1 问题描述	167
11.3.2 使用适配器模式实现电子商务系统功能的扩展	168
11.3.3 适配器模式的应用效果分析	174
11.4 本章习题	175

第12章 享元模式	177
-----------	-----

12.1 享元模式概述	177
12.1.1 享元模式的动机	177

12.1.2	享元模式的定义	178
12.2	享元模式的结构及其代码示例	178
12.2.1	单纯享元模式的结构	178
12.2.2	单纯享元模式的代码示例——文本编辑器中的应用	179
12.2.3	复合享元模式的结构	181
12.2.4	复合享元模式的代码示例——文本编辑器中的应用	183
12.3	享元模式在电子商务系统中的应用	186
12.3.1	问题描述	186
12.3.2	使用享元模式实现淘宝网网店模板的共享	186
12.3.3	享元模式的应用效果分析	189
12.4	本章习题	190
第 13 章	代理模式	192
13.1	代理模式概述	192
13.1.1	代理模式的动机	192
13.1.2	代理模式的定义	192
13.2	代理模式的结构及其代码示例	193
13.2.1	代理模式的结构	193
13.2.2	代理模式的代码示例——票务中心代购车票	194
13.3	代理模式在电子商务系统中的应用	196
13.3.1	问题描述	196
13.3.2	远程代理示例——商品信息查询 Web Service 服务器端及客户端的实现	197
13.3.3	代理模式的应用效果分析	203
13.4	本章习题	203
第 14 章	桥接模式	205
14.1	桥接模式概述	205
14.1.1	桥接模式的动机	205
14.1.2	桥接模式的定义	206
14.2	桥接模式的结构及其代码示例	206
14.2.1	桥接模式的结构	206
14.2.2	桥接模式的代码示例——日志记录工具的开发	207
14.3	桥接模式在电子商务系统中的应用	213
14.3.1	问题描述	213
14.3.2	使用桥接模式实现电子商务系统中的发送消息通知的功能	213
14.3.3	桥接模式的应用效果分析	217
14.4	本章习题	218

第 15 章 组合模式	221
15.1 组合模式概述	221
15.1.1 组合模式的动机	222
15.1.2 组合模式的定义	222
15.2 组合模式的结构及其代码示例	222
15.2.1 组合模式的结构	222
15.2.2 组合模式的代码示例——二叉树遍历	223
15.3 组合模式在电子商务系统中的应用	228
15.3.1 问题描述	228
15.3.2 使用组合模式管理网站商品类别	228
15.3.3 组合模式的应用效果分析	234
15.4 本章习题	235
第 16 章 结构型设计模式总结	238
16.1 结构型模式小结	238
16.2 结构型模式综合练习	239
第四部分 行为型设计模式	
第 17 章 观察者模式	251
17.1 观察者模式概述	251
17.1.1 观察者模式的动机	251
17.1.2 观察者模式的定义	251
17.2 观察者模式的结构及其代码示例	252
17.2.1 观察者模式的结构	252
17.2.2 观察者模式的代码示例——微信公众平台服务系统模拟	253
17.3 观察者模式在电子商务系统中的应用	259
17.3.1 问题描述	259
17.3.2 使用观察者模式实现电子商务系统中的商品订阅通知	259
17.3.3 观察者模式的应用效果分析	264
17.4 本章习题	265
第 18 章 策略模式	267
18.1 策略模式概述	267
18.1.1 策略模式的动机	267
18.1.2 策略模式的定义	268
18.2 策略模式的结构及其代码示例	268
18.2.1 策略模式的结构	268

18.2.2	策略模式的代码示例——Oracle 数据库备份策略模拟	269
18.3	策略模式在电子商务系统中的应用	271
18.3.1	问题描述	271
18.3.2	使用策略模式实现商品排序功能	272
18.3.3	策略模式的应用效果分析	280
18.4	本章习题	281
第 19 章	责任链模式	283
19.1	责任链模式概述	283
19.1.1	责任链模式的动机	283
19.1.2	责任链模式的定义	284
19.2	责任链模式的结构及其代码示例	284
19.2.1	责任链模式的结构	284
19.2.2	责任链模式的代码示例——聚餐费用申请管理	285
19.3	责任链模式在电子商务系统中的应用	290
19.3.1	问题描述	290
19.3.2	使用责任链模式实现电子商务系统中的敏感词过滤	290
19.3.3	责任链模式的应用效果分析	294
19.4	本章习题	295
第 20 章	状态模式	297
20.1	状态模式概述	297
20.1.1	状态模式的动机	297
20.1.2	状态模式的定义	297
20.2	状态模式的结构及其代码示例	298
20.2.1	状态模式的结构	298
20.2.2	状态模式的代码示例——电灯的开关控制	298
20.3	状态模式在电子商务系统中的应用	300
20.3.1	问题描述	300
20.3.2	使用状态模式实现电子商务系统中会员等级状态的转换	301
20.3.3	状态模式的应用效果分析	305
20.4	本章习题	306
第 21 章	命令模式	311
21.1	命令模式概述	311
21.1.1	命令模式的动机	311
21.1.2	命令模式的定义	311
21.2	命令模式的结构及其代码示例	312
21.2.1	命令模式的结构	312

21.2.2	命令模式的代码示例——计算机启动过程模拟	313
21.3	命令模式在电子商务系统中的应用	316
21.3.1	问题描述	316
21.3.2	使用命令模式实现电子商务系统中的用户管理	317
21.3.3	命令模式的应用效果分析	324
21.4	本章习题	325
第 22 章 中介者模式		327
22.1	中介者模式概述	327
22.1.1	中介者模式的动机	327
22.1.2	中介者模式的定义	328
22.2	中介者模式的结构及其代码示例	328
22.2.1	中介者模式的结构	328
22.2.2	中介者模式的代码示例——房屋中介模拟	329
22.3	中介者模式在电子商务系统中的应用	332
22.3.1	问题描述	332
22.3.2	使用中介者模式模拟当当网的送货流程	333
22.3.3	中介者模式的应用效果分析	338
22.4	本章习题	339
第 23 章 行为型设计模式总结		341
23.1	行为型模式小结	341
23.2	行为型模式综合练习	344
附录 习题答案		350
参考文献		362

：新得台聯本

第一 部分

设计模式基础

第1章 设计模式概述

第2章 设计模式与UML建模

第3章 面向对象的设计原则

本部分导读：

设计模式描述了软件设计过程中某一类常见问题的一般性的解决方案。

面向对象设计模式描述了面向对象设计过程中、特定场景下、类与相互通信的对象之间常见的组织关系。

人们通常所说的设计模式隐含地表示“面向对象设计模式”，但这并不意味着“设计模式”就等于“面向对象设计模式”。除了“面向对象设计模式”外，还有其他设计模式。除了前人已总结过的设计模式外，还有更多的面向对象设计模式。

本书中的设计模式即指“面向对象设计模式”，与算法技巧不同，面向对象设计模式不可以照搬照用，它是建立在对“面向对象”深入理解的基础上的经验性知识。

本部分主要介绍设计模式的起源及基本概念、面向对象设计模式与 UML 建模、面向对象的设计原则。

第1章

设计模式概述

本章主要介绍设计模式的基本概念、为什么要使用模式、如何正确地理解和使用设计模式；本章还对 GoF 的 23 种设计模式做了简要介绍，对设计模式、架构、框架等易混淆概念进行了比较。

1.1 设计模式的定义与起源

1.1.1 模式与设计模式

模式(Pattern)是指从生产经验和生活经验中经过抽象和升华提炼出来的核心知识体系,也可看作是解决某一类问题的方法。把解决某类问题的方法总结归纳到理论高度,就是模式。模式是一种指导,这种捕获有效技术的思想可以应用到很多领域中,如烹饪、产品生产、软件开发及其他一些行业。模式是集体智慧的结晶,它概括了相关行业中经验丰富的从业者所应用的所有概念和方法。

对于一个成熟的模式,它作为一种问题的解决思路,往往已经适用于一个实践环境,并且可以适用于其他环境。通常来说,模式具有以下特点:

- (1) 在特定的场景下有可重用性,对相同类型不同问题的环境,其解决方案都有效。
- (2) 可传授性,即问题出现的机会很多,解决问题的方案相同,人们相对可以接受。
- (3) 有表示模式的名称。

“设计模式(Design pattern)”也是一种模式,是一套被反复使用、多数人知晓的、经过分类编目的、代码设计经验的总结。使用设计模式是为了可重用代码,让代码更容易被他人理解,保证代码的可靠性。

1.1.2 模式的起源

模式起源于建筑业而非软件业。Christopher Alexander(克里斯托弗·亚历山大)最早提出“设计模式”的概念。他是哈佛大学的建筑学博士、美国加州大学伯克利分校建筑学教授、加州大学伯克利分校环境结构研究所所长、美国艺术和科学院院士,人称模式之父。

Christopher Alexander 博士及其研究团队用了约 20 年的时间,对住宅和周边环境进行了大量的调查研究和资料收集工作,发现人们对舒适住宅和城市环境存在一些共同的认同

规律,在1970年他的《建筑的永恒之道》里描述了设计模式的发现,其著作 *A Pattern Language: Towns, Buildings, Construction* (Alexander 等人编著,牛津大学出版社于1977年出版)介绍了250多种建筑设计模式,对每一个模式(Pattern)都从 Context(前提条件)、Theme 或 Problem(目标问题)、Solution(解决方案)三个方面进行了描述,并给出了从用户需求分析到建筑环境结构设计直至经典实例的过程模型。该著作影响深远,尽管它是针对建筑领域的,但他的观点实际上适用于所有的工程设计领域,其中也包括软件设计领域。

1990年,软件工程界开始关注 Christopher Alexander 等这一在住宅、公共建筑与城市规划领域的重大突破。最早将模式的思想引入软件工程方法学的是1991—1992年以“四人组(Gang of Four,简称 GoF,分别是 Erich Gamma、Richard Helm、Ralph Johnson 和 John Vlissides)”自称的四位著名软件工程学者,他们在1994年归纳发表了23种在软件开发中使用频率较高的设计模式,旨在用模式来统一沟通面向对象方法在分析、设计和实现间的鸿沟。1995年,GoF 将收集和整理好的23种设计模式汇编成 *Design Patterns: Elements of Reusable Object-Oriented Software* (《设计模式:可复用面向对象软件的基础》)一书,该书的出版标志着设计模式正式成为面向对象(Object Oriented)软件工程的一个重要研究分支。软件设计模式是针对某一类问题从许多优秀的软件系统中总结出的成功且可复用的设计方案,是针对某一类问题的最佳设计解决方案,是面向对象语言中利用类和方法实现编程目标的解决方案。

需要明确的是,软件模式并非仅限于设计模式,还包括架构模式、分析模式和过程模式等,实际上,在软件开发生命周期的每一个阶段都存在着一些被认同的模式。软件模式与具体的应用领域无关,也就是说,无论你从事的是移动应用开发、桌面应用开发、Web 应用开发还是嵌入式软件的开发,都可以使用软件模式。

1.1.3 设计模式的基本要素

在面向对象软件开发领域,常说的设计模式主要指 GoF 在《设计模式:可复用面向对象软件的基础》一书中所介绍的23种经典设计模式。与 GoF 模式相对应的另一种重要的设计模式是通用责任分配软件模式,即 GRASP 模式,其英文全称为 General Responsibility Assignment Software Pattern。GRASP 模式站在面向对象设计的角度,告诉我们怎样设计问题空间中的类与分配它们的行为职责,以及明确类之间的相互关系等。GRASP 模式着重考虑设计类的原则及如何分配类的功能,指导该由谁来创建类;而 GoF 模式则着重考虑设计的实现、类的交互及软件质量。可以说,GoF 模式就是符合 GRASP 模式要求的面向对象设计模式。

设计模式并不仅仅只有 GoF 归纳的23种,随着软件开发技术的发展,越来越多的新模式不断诞生并得以应用。对于一个新的设计模式,它必须满足大三律(Rule of Three),意即:一个解决方案必须在三个完全不同类型或者不同领域的系统中得以应用才有资格成为候选模式。

设计模式使人们可以更加简单方便地复用成功的设计和体系结构,而将已证实的技术

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。描述一个设计模式通常包含下列四种基本要素：

1. 模式名称

模式名称(pattern name)是一个助记名,它用一两个词来描述模式的问题、解决方案和效果。命名一个新的模式将增加我们的设计词汇,基于一个模式词汇表,同事之间就可以讨论模式并在编写文档时使用它们。模式名可以帮助我们思考,便于我们与其他人交流设计思想及设计结果。找到恰当的模式名也是设计模式编目工作的难点之一。

2. 问题

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。

3. 解决方案

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。因为模式就像一个模板,可应用于多种不同场合,所以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。

4. 效果

效果(consequences)描述了模式应用的效果及使用模式应权衡的问题。尽管我们描述设计决策时,并不总提到模式效果,但它们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,它们也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,所以模式效果包括它对系统的灵活性、扩充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很有帮助。

例如,观察者模式:

- 名称——观察者。
- 问题——定义对象间的一种一对多的依赖关系,当一个对象的状态发生改变时,所有依赖于它的对象都得到通知并被自动更新。
- 解决方案——主题(Subject)接口、观察者(Observer)接口、具体主题(ConcretSubject)、具体观察者(ConcretObserver)。
- 效果——一个对象(目标对象)的状态发生改变,所有的依赖对象(观察者对象)都将得到通知,将被观察者和观察者解耦,保证高度的协作。

其模式结构图如图 1.1 所示。

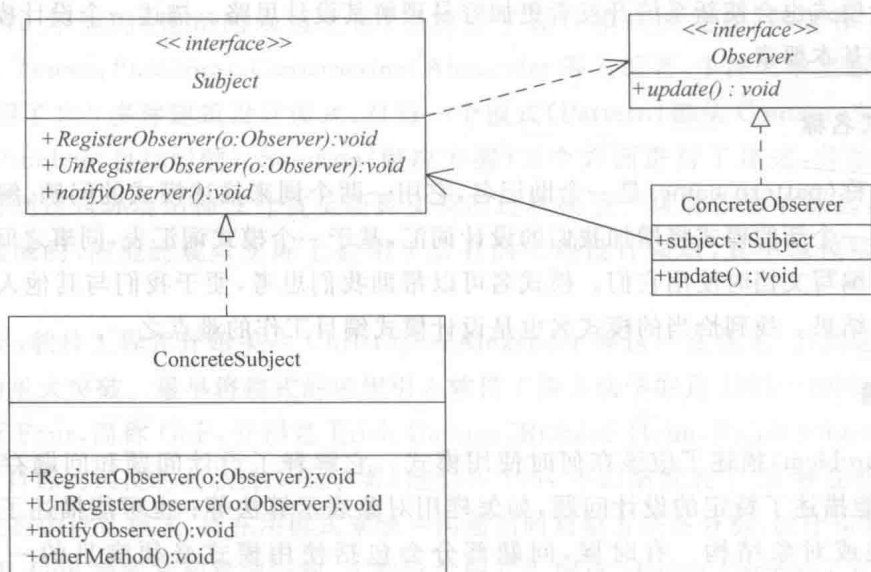


图 1.1 观察者模式类图

1.2 为什么要使用设计模式

1.2.1 设计模式的作用

如今,设计模式是面向对象编程的热门话题之一,越来越多的开发人员认识到设计模式的重要性。从1995年至今,无论是在大型API或框架(如JDK、.NET Framework等)、轻量级框架(如Struts、Spring、Hibernate、JUnit等),还是在应用软件的开发中,设计模式都得到了广泛的应用。设计模式是面向对象设计过程中常见问题的比较成熟的解决方案,是前辈们在程序设计过程中总结、抽象出来的通用优秀经验,学好并正确使用设计模式对软件项目开发意义重大。

(1) 设计模式以一种标准的方式供广大开发人员使用,为开发者的沟通提供了一套机制,帮助开发者更好地明白和更清晰地描述一段被给出的代码。例如,比较如下甲乙二人对订单的描述。

甲:我要一份涂了奶酪以及果酱的白面包、加了香草冰淇淋的巧克力汽水、夹了培根的火烤起司三明治、鲑鱼色拉吐司、香蕉船(有冰淇淋和香蕉片)、一杯加了奶精和两颗糖的咖啡还有一个烧烤汉堡!

乙:给我一份C.J.白的,一个黑与白,一份杰克班尼,一份Radio,一份主厨船,一个普通咖啡,还有给我烧一个!

显然,同样一份订单,由于乙和厨师之间有“共享词汇”,甲不懂这些词汇,使得甲讲话的长度多了一倍。共享词汇方便顾客点餐,方便顾客与厨师间的沟通。

设计模式同样让开发人员之间有一套“共享词汇”,一旦懂这些词汇,开发人员之间沟通这些观念就很容易,也会促使那些不懂的程序员想开始学习设计模式。