

本书帮您圆满实现 快速递交高质量软件产品之梦!

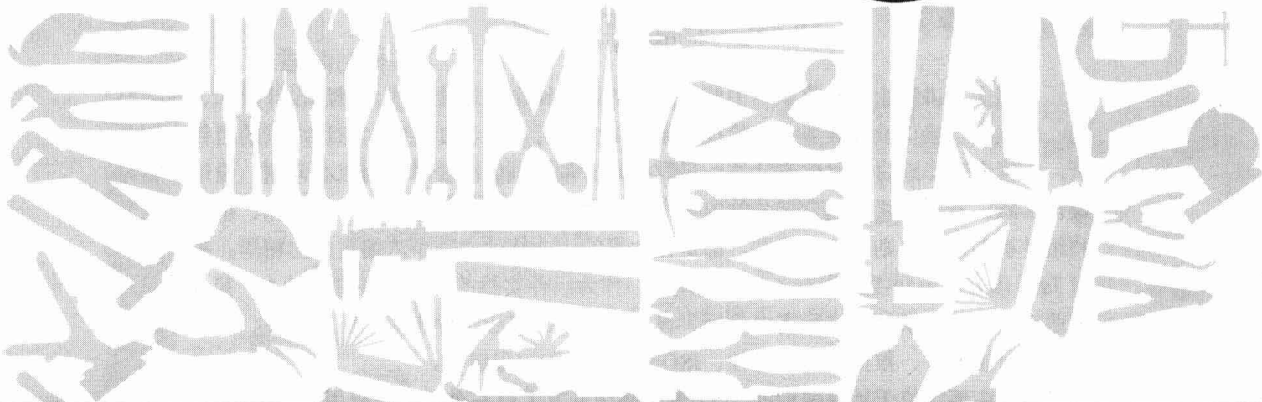
Java 组件设计

孔德生 编著

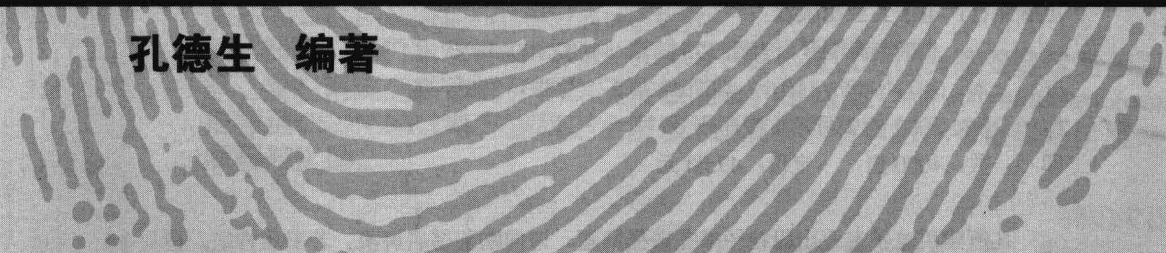


电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Java技术大系



Java 组件设计



孔德生 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书主要讲解企业应用系统中核心组件的设计原则与最佳实践，澄清设计模式、数据结构、多线程、接口设计等多个高阶技术领域“流行”的认知误区，通过大量的组件实例分析，为读者精彩地讲解组件设计这一最具技术含量的领域需要考虑的核心问题、设计方案与最佳实践。本书主要涵盖三部分的内容：第1篇，主要讲解了应用程序的架构、组件的定义和组件核心设计原则。这些原则，都是在大量的企业软件实践中浓缩提炼的精华；第2篇，对Java语言的高级主题进行了详尽的阐述，作为组件实战的必需必备基础功力；第3篇，对企业应用的核心组件，进行需求分析、设计和实现。核心组件包括：配置组件、Socket通信组件、日志组件、数据库访问组件、JSON转换器和权限组件。

本书主要定位于软件架构师、设计师、高级开发人员在企业实际应用开发中的参考读物，也适合大专院校相关专业作为教材辅导图书阅读。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

Java 组件设计 / 孔德生编著. —北京：电子工业出版社，2009.4
（Java 技术大系）
ISBN 978-7-121-08165-1

I. J… II. 孔… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字（2009）第 009255 号

责任编辑：孙学瑛

印 刷：北京智力达印刷有限公司

装 订：北京中新伟业印刷有限公司

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：787×980 1/16 印张：20.5 字数：370 千字

印 次：2009 年 4 月第 1 次印刷

印 数：4000 册 定价：49.00 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至 dbqq@phei.com.cn。

服务热线：（010）88258888。



前言

设计，决定了软件质量！

组件化设计和构建软件系统，是实现快速发布高质量软件产品之梦的最佳实践！

组件设计，凝聚了需求分析、面向对象、设计模式、数据结构、多线程等一系列高阶领域的核心精髓！

Java 语言，是互联网上的卓越语言，从诞生的那一天开始，就被很多软件工程师所青睐，不断在世界的每个角落开花结果，应用日益广泛。

Java 开源世界，如火如荼地发展，无数的框架、组件如雨后春笋般涌现，极大地推进了 Java 技术在各个领域的商业应用。

然而，由于设计者的眼界、经验和水平等的限制，Java 开源框架、组件参差不齐，优秀者并不多见。在企业软件开发中，Java 往往成了臃肿、笨拙、低效等的代名词。功能最简单的 Java 组件也要几十、上百个类文件，使用组件需要引入的 Jar 文件少则几兆字节，多则几十兆、上百兆字节。因此，所谓鼓吹的轻量级组件，不过是个噱头而已。

笔者从事企业软件开发十余载，专注平台和组件开发领域，深知组件设计与开发过程的“高处不胜寒”。因此，在本书中，笔者将自己对组件技术的认识和心得进行总结和提炼，为读者奉上一份 Java 组件设计领域的精彩技术大餐。

本书定位

本书面向软件架构师、设计师、高级开发人员，讲解企业应用中核心组件的设计原则与最佳实践。

本书将澄清设计模式、数据结构、多线程、接口设计等多个高阶技术领域“流行”的认知误区，通过大量的组件实例分析，为读者精彩讲解组件设计这一最具技术含量的领域需要考虑的核心问题、设计方案与最佳实践。

本书的内容

这本书主要涵盖三部分的内容：

第 1 篇，主要讲解了应用程序的架构、组件的定义和组件核心设计原则。这些原则，都是在大量的企业软件实践中浓缩提炼的精华。

第 2 篇，对 Java 语言的高级主题进行了详尽的阐述，作为组件实战的必需必备基础功力。

第 3 篇，对企业应用的核心组件，进行需求分析、设计和实现。核心组件包括：

- 第 4 章，配置组件
- 第 5 章，Socket 通信组件
- 第 6 章，日志组件
- 第 7 章，数据库访问组件
- 第 8 章，JSON 转换器
- 第 9 章，权限组件

本书代码下载地址为 <http://www.broadview.com.cn>。

特别致谢

衷心感谢周坚先生，在华为公司期间对我无私的指导，使我有幸步入设计领域的高级殿堂！

衷心感谢我的父亲、母亲和兄弟姐妹，是你们的关怀照顾和言传身教，让我懂得人生的意义！

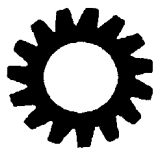
衷心感谢我的妻子，默默忍受我长期的熬夜，不断给我关心、支持和鼓励！

衷心感谢我的儿子，给我带来无尽的欢乐！

衷心感谢在我的生活、工作中，帮助过我的每一个人！

孔德生

2009 年 2 月



目录

第 1 篇 组件设计总括

第 1 章 组件设计概述	2
1.1 应用软件的总体架构	2
1.2 组件定义及与其相关概念的澄清	4
1.3 总结	7
第 2 章 组件设计原则	8
2.1 组件定位：精准地解决共性问题	10
2.2 组件设计：无配置文件	11
2.3 组件设计：与使用者概念一致	12
2.4 组件设计：业务无关的中立性	13
2.5 组件设计实现：对使用环境无假设	13
2.6 组件设计实现：单类设计和实现	14
2.7 总结	14

第 2 篇 组件设计的基础知识

第 3 章 预备知识	16
3.1 Java 语法深入讲解	16
3.1.1 static	16
3.1.2 嵌套类	18
3.1.3 静态嵌套类	20
3.2 反射	21
3.3 数据结构	23
3.3.1 ArrayList	23
3.3.2 LinkedList	24
3.3.3 HashSet	24
3.3.4 HashMap	25
3.4 泛型	26
3.4.1 泛型简介	26
3.4.2 泛型应用在接口上	30
3.4.3 泛型应用在类上	30
3.4.4 泛型应用在方法上	31
3.4.5 限制泛型的可用类型	31
3.4.6 通配符泛型	34
3.4.7 泛型深入	36
3.5 线程	42
3.5.1 线程基础	42
3.5.2 多线程同步	46
3.5.3 线程的阻塞	50

3.5.4 守护线程	52
3.5.5 多线程范例	52

第 3 篇 组件设计

第 4 章 配置组件	61
------------------	----

4.1 配置文件格式	61
4.2 DTD, 还是 Schema.....	64
4.3 接口设计	68
4.3.1 设计思路	68
4.3.2 复合元素映射.....	69
4.3.3 接口设计	73
4.3.4 接口重构	78
4.4 接口实现	85
4.4.1 主要实现结构.....	85
4.4.2 DOM 根节点解析	86
4.4.3 XML 数据设置为对象属性.....	87
4.4.4 补充说明	92
4.5 总结	92

第 5 章 Socket 通信组件	93
-------------------------	----

5.1 使用场景	93
5.2 需求分析	93
5.2.1 系统内, 还是系统间.....	94
5.2.2 TCP, 还是 UDP	94
5.2.3 点对点, 还是广播.....	95

5.2.4	单连接，还是多连接.....	95
5.2.5	安全问题.....	96
5.2.6	包边界问题.....	96
5.2.7	连接管理.....	97
5.3	客户端接口设计.....	99
5.3.1	设计原则.....	99
5.3.2	配置接口.....	100
5.3.3	功能接口.....	101
5.3.4	事件通知接口.....	101
5.4	客户端技术实现.....	104
5.4.1	技术实现选型.....	104
5.4.2	配置接口实现.....	104
5.4.3	功能接口实现.....	104
5.5	服务端接口设计.....	113
5.5.1	设计原则.....	113
5.5.2	配置接口.....	114
5.5.3	功能接口.....	115
5.5.4	事件通知接口.....	116
5.6	服务端技术实现.....	118
5.6.1	技术实现选型.....	118
5.6.2	配置接口实现.....	119
5.6.3	功能接口实现.....	120
5.7	总结.....	133
第 6 章	日志组件	134
6.1	使用场景.....	134

6.2	需求分析	137
6.2.1	单例，或多例	137
6.2.2	异步，还是同步	137
6.2.3	单线程，还是多线程	137
6.2.4	集中监控	138
6.2.5	在线修改配置	138
6.2.6	日志格式	139
6.2.7	日志内容	139
6.2.8	容错考虑	140
6.3	接口设计	141
6.3.1	核心设计要点	141
6.3.2	配置接口设计	141
6.3.3	功能接口设计	142
6.3.4	日志转换接口	144
6.4	接口实现	145
6.4.1	技术选型	145
6.4.2	配置接口实现	145
6.4.3	功能接口实现	148
6.4.4	日志转换接口实现	153
6.5	总结	153
第 7 章	数据库访问组件	154
7.1	需求分析	155
7.1.1	数据库类型	155
7.1.2	数据库连接	155
7.1.3	单实例，还是多实例	156

7.1.4	ORM, 还是 SQL	156
7.1.5	什么样的结果集	156
7.1.6	结果集分页、合并	157
7.1.7	存储过程	157
7.1.8	对象的直接数据操作	158
7.1.9	事务	159
7.1.10	缓存	159
7.2	DBAccess 接口设计	160
7.2.1	设计思路	160
7.2.2	配置接口设计	163
7.2.3	连接管理	164
7.2.4	SQL 接口设计	166
7.2.5	批操作接口	169
7.2.6	存储过程接口设计	169
7.2.7	事务接口	178
7.2.8	结果集转换为对象	179
7.2.9	对象的直接数据库操作	180
7.2.10	接口汇总	181
7.3	DBMonitor 设计	186
7.4	数据对象映射规则设计	189
7.4.1	规则表示法	189
7.4.2	定义 Annotation	190
7.4.3	代码生成工具	193
7.5	DataSet 设计	199
7.5.1	基本信息	200



7.5.2	数据的随机访问.....	200
7.5.3	列值读取	201
7.5.4	数据的增、删、改.....	202
7.5.5	数据集的分割、合并.....	203
7.5.6	DataSetMetaData	204
7.5.7	DataSet 接口汇总	205
7.6	DBAccess 接口实现.....	207
7.6.1	配置接口实现.....	207
7.6.2	连接管理实现.....	208
7.6.3	事务接口实现.....	214
7.6.4	SQL 接口实现	216
7.6.5	批操作接口实现.....	222
7.6.6	存储过程接口实现.....	223
7.6.7	数据集转换为对象实现.....	229
7.6.8	对象的直接数据库操作实现.....	235
7.7	总结	237
第 8 章	JSON 转换器	238
8.1	引言	238
8.1.1	传统 Web 应用	238
8.1.2	Ajax 应用	239
8.1.3	数据规则	241
8.2	需求分析	243
8.2.1	需求场景	243
8.2.2	核心功能	244
8.3	接口设计	247

8.3.1	设计原则	247
8.3.2	JSON->Java	248
8.3.3	Java->JSON	248
8.3.4	接口汇总	249
8.4	接口实现	249
8.4.1	json2Obj 实现	249
8.4.2	obj2Json 实现	254
8.5	总结	257
第 9 章	权限组件	258
9.1	引言	258
9.2	需求分析	258
9.2.1	应用场景	258
9.2.2	权限分类	260
9.2.3	角色	265
9.2.4	组织结构	266
9.2.5	需求汇总	266
9.3	接口设计	267
9.3.1	标识用户	268
9.3.2	获得发起请求的用户	268
9.3.3	拦截 HTTP 请求	270
9.3.4	权限组件和应用系统之间访问	270
9.3.5	应用系统的权限定义	273
9.3.6	应用系统调用权限组件	282
9.3.7	用户权限变更	282
9.3.8	Session 超时失效	283

9.3.9	AuthFilter 接口汇总	285
9.3.10	AuthFilterImpl 实现原型	286
9.3.11	部署配置	287
9.4	接口实现	288
9.4.1	实现思路	288
9.4.2	私有成员变量	288
9.4.3	init 方法实现	290
9.4.4	doFilter 方法实现	293
9.4.5	canAccess 方法实现	298
9.4.6	reloadRoleAuths 方法实现	298
9.4.7	sessionDestroy 方法实现	300
9.5	权限应用示例	301
9.5.1	系统简介	301
9.5.2	运行示例	302
9.5.3	实现介绍	305
9.6	总结	308



重点关注

⚙	共性的业务问题，是组件的产生来源。.....	10
⚙	技术与业务对齐，这是组件设计的第一位重要原则。.....	11
⚙	最简单的组件，就是一个类。.....	14
⚙	高内聚！高内聚！单类组件，简单到极致！.....	14
⚙	本质上讲，static 关键字声明了一个全局变量。.....	16
⚙	在泛型代码中，根本就没有参数化类型的信息。.....	39
⚙	发布函数的原型相对稳定，不会频繁发生修改。.....	76
⚙	从接口上看，只有一个 Static 方法，没有任何 Get 或 Set 方法，因此不满足 IOC 容器的规范，无法被放入 IOC 容器中，因此也无法与其他组件进行基于 IOC 容器的集成。这可是个大问题!!	83
⚙	因此，使用 IOC 容器作为物理部署、实施的配置方式，是不合适的。这里编写的配置组件，是用于部署实施的有力武器！.....	83

- ⚙ 获取满足条件的 Key 集合，遍历 Key 集合，取出每个 Key，判断 Key 是否是下面三种状态之一：acceptable，readable，writable，从而对应地进行接收连接、数据读取、发送数据的操作。 127

- ⚙ 大量的软件开发实践证明，进行复杂数据操作的应用系统中，存储过程是解决问题的最有效方案。 158

- ⚙ 应用程序的开发者，不需要编写太多类，就可以将结果集转换为对象，获得类似 ORM 框架的功能。最理想的情况就是一个类也不增加。 179

- ⚙ 作为组件设计者，应该让组件的使用者做且仅做他们必须做的事情，摒弃任何不必要、烦琐、无意义的环节。 189

- ⚙ 这种情况下，如何实现互相之间的访问，就成了一个难题！ 271

- ⚙ 大多数 Web 应用，使用 Spring 作为 IOC 容器来实例化、配置各种后台组件。而 Spring 自身在 Deployment Descriptor 中是作为一个 Listener 被 Web Container 实例化的。Listener 的初始化过程，早于 Filter 的初始化过程，因此当权限组件开始被 Web Container 初始化时，Spring IOC 容器和 IOC 容器内的各种后台组件都已经初始化完毕。这样，在权限组件的初始化参数中，配置一个类名，应用程序提供这个类的实现，而权限组件动态实例化这个类，通过这个类的实例作为桥梁，获得应用程序实例的访问，这种方式是可行的。 271

- ⚙ 只有角色产生了变化，才需要权限组件更新内部的权限数据。而用户变化、用户权限、用户角色变化等，都不需要权限组件更新内部的权限数据。 282



Java 组件设计

第 1 篇 组件设计总括

第 1 章 组件设计概述..... 1

面对复杂而日益多变的客户需求，如何应对？如何快速产出高质量的软件？经过多年的软件工程实践，组件技术被认为是快速发布高质量软件产品的关键技术。通过一系列基础组件的组装，可以像搭积木一样，迅速生成所需软件的主体结构和核心功能，应用程序开发者只需要编写特定的应用逻辑即可，这样大大加快了软件产品的开发速度。另外，组件都是经过了充分测试的，因此在质量上也提供了坚实的保障。只有这样，才能实现快速、高质量的软件开发。

第 2 章 组件设计原则..... 7

本章将对目前业界存在的诸多技术争论、误区进行澄清，让读者从所谓的“业界潮流”、教条、“黄金定律”中走出来，真正理解组件设计过程的核心原则。