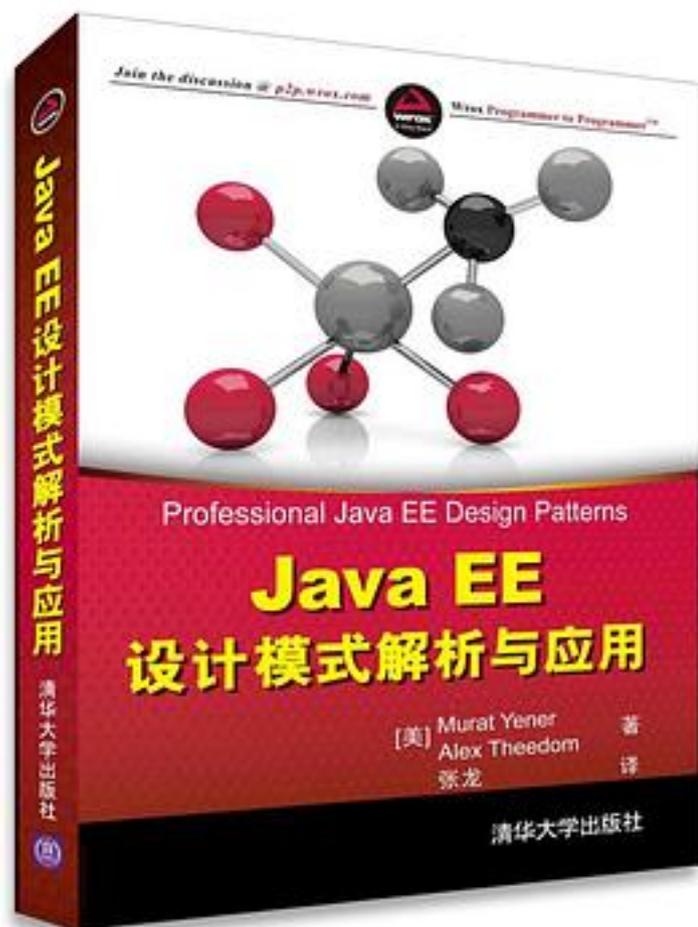


Java EE设计模式解析与应用



[Java EE设计模式解析与应用_下载链接1](#)

著者:(美) 叶尼(Yener, M.), (美) 希多姆(Theedom, A.) 著

出版者:清华大学出版社

出版时间:2015-10-1

装帧:平装

isbn:9787302415862

用Java EE设计模式解决实际问题

模式是Java开发者必需的设计工具。虽然Java EE提供了许多“经典GoF模式”的默认实现，但大多数富有经验的开发者都低估了这些开箱即用的实现的价值。《Java EE设计模式解析与应用》一书分析每种模式的意图，并通过各种可供选择的示例代码演示其实现，从而帮助广大开发者改进代码质量并提升架构设计水平。

本书首先呈现最常见的GoF模式，接着循序渐进地讲解新模式；将重温这些模式的经典实现并在示例代码的引导下透彻分析Java EE特定的实现方式。为紧贴实际讲解这些模式，作者还讲述了自己亲历的成功和失败项目，这些经历有些十分有趣，有些令人沮丧，但都很好地阐释了在实际中如何灵活运用这些原理。

主要内容

- ◆ 学习Java EE如何通过注解以及比J2EE更整洁的编程模型提供诸多模式的实现
- ◆ 用Java EE实现重要模式，包括门面、单例、工厂、装饰、观察者、依赖注入以及针对Web应用的MVC模式
- ◆ 掌握Java EE编码方法，包括面向方面编程、异步编程以及REST风格的Web Service
- ◆ 学习模式运用不当的后果以及如何避免常见陷阱
- ◆ 学习避用模式的时机和原因，并给出替代方案

作者介绍:

Murat

Yener是一位Android极客，就职于Intel，负责开发桌面应用及HTML5移动应用。Murat具有丰富的Java、Web、Java EE和OSGi开发经验，是Eclipse提交者和GDG Istanbul用户组管理者，曾多次在重要技术会议上发表演讲。

Alex Theedom是Indigo Code Collective(E-scape

group)的资深开发者，具有丰富的Java EE和Spring Web应用开发经验。Alex曾为欧洲多个组织开发基于微服务的架构、ATM软件以及在线学习系统。

目录: 目录

第 I 部分 Java EE设计模式介绍

第1章 设计模式概览 3

1.1 何为设计模式 4

1.1.1 如何检测模式，为何需要模式 6

1.1.2 现实世界中的模式 6

1.2 设计模式基础 8

1.3 企业模式 8

1.3.1 从Java到企业级Java 9

1.3.2 企业Java模式的出现 10

1.3.3 设计模式与企业模式 11

1.3.4 当传统设计模式遇到Java EE 12

1.3.5 当模式变成反模式 13

1.4 小结 14

1.5 相关资料	14
第2章 Java EE基础	17
2.1 多层架构	19
2.2 客户端层	20
2.3 中间层	21
2.3.1 Web层	21
2.3.2 业务层	22
2.4 EIS层	23
2.5 Java EE服务器	24
2.6 Java EE Web Profile	25
2.7 Java EE核心原则	25
2.8 约定优于配置	26
2.9 上下文与依赖注入	26
2.10 拦截器	28
2.11 小结	30
2.12 本章练习	30
第II部分 实现Java EE设计模式	
第3章 门面模式	33
3.1 何为门面	34
3.2 使用普通代码实现门面模式	36
3.3 使用Java EE实现门面模式	38
3.3.1 使用无状态Bean实现门面	38
3.3.2 使用有状态Bean实现门面	41
3.4 何时以及何处该使用门面模式	42
3.5 小结	43
3.6 本章练习	43
3.7 相关资料	43
第4章 单例模式	45
4.1 何为单例?	46
4.1.1 单例模式类图	47
4.1.2 使用普通代码实现单例模式	48
4.2 使用Java EE实现单例模式	53
4.2.1 单例Bean	53
4.2.2 在启动时使用单例	54
4.2.3 确定启动顺序	56
4.2.4 管理并发	59
4.3 何处以及何时该使用单例模式	63
4.4 小结	63
4.5 本章练习	64
4.6 相关资料	65
第5章 依赖注入与CDI	67
5.1 何为依赖注入?	68
5.2 使用普通代码实现DI	69
5.3 使用Java EE实现DI	73
5.3.1 @Named注解	75
5.3.2 上下文与依赖注入(CDI)	76
5.3.3 CDI与EJB	77
5.3.4 CDI Bean	77
5.3.5 @Inject注解	79
5.3.6 上下文与作用域	79
5.3.7 命名与EL	80
5.3.8 用于Backing JSF的CDI Bean	81
5.3.9 限定符	81
5.3.10 Alternatives	82

5.3.11	Stereotypes	83
5.3.12	通过CDI实现的其他模式	84
5.4	小结	84
5.5	本章练习	85
5.6	相关资料	85
第6章	工厂模式	87
6.1	何为工厂?	88
6.2	工厂方法	89
6.3	抽象工厂	94
6.4	使用Java EE实现工厂模式	96
6.5	何处以及何时该使用工厂模式	110
6.6	小结	111
6.7	本章练习	111
6.8	相关资料	111
第7章	装饰模式	113
7.1	何为装饰器?	114
7.2	使用普通代码实现装饰模式	116
7.3	使用Java EE实现装饰模式	121
7.4	何处以及何时该使用装饰模式	129
7.5	小结	130
7.6	本章练习	130
7.7	相关资料	131
第8章	面向方面编程(拦截器)	133
8.1	何为面向方面编程?	134
8.2	使用普通代码实现AOP	137
8.3	Java EE中的方面——拦截器	140
8.3.1	拦截器生命周期	145
8.3.2	默认级别的拦截器	146
8.3.3	拦截器顺序	147
8.3.4	CDI拦截器	150
8.4	何处以及何时该使用拦截器	153
8.5	小结	154
8.6	相关资料	155
第9章	异步	157
9.1	何为异步编程	158
9.2	使用普通代码实现异步模式	161
9.3	Java EE中的异步编程	164
9.3.1	异步Bean	164
9.3.2	异步Servlet	167
9.4	何处以及何时该使用异步编程	172
9.5	小结	174
9.6	本章练习	174
9.7	相关资料	175
第10章	定时器服务	177
10.1	何为定时器服务	178
10.2	使用Java EE实现定时器	181
10.2.1	自动化定时器	181
10.2.2	程式化定时器	183
10.2.3	定时器表达式	187
10.2.4	事务	190
10.3	小结	191
10.4	本章练习	192
10.5	相关资料	192
第11章	观察者模式	193

- 11.1 何为观察者? 194
 - 11.1.1 说明 195
 - 11.1.2 观察者类图 197
- 11.2 使用普通代码实现观察者模式 197
- 11.3 使用Java EE实现观察者模式 200
- 11.4 何处以及何时该使用观察者模式 209
- 11.5 小结 210
- 11.6 本章练习 211
- 11.7 相关资料 211
- 第12章 数据访问模式 213
 - 12.1 何为数据访问模式? 214
 - 12.2 数据访问模式概览 216
 - 12.2.1 数据传输对象模式 216
 - 12.2.2 Java持久化架构API与对象关系映射 217
 - 12.3 使用Java EE实现数据访问模式 218
 - 12.4 何处以及何时该使用数据访问模式 227
 - 12.5 小结 227
 - 12.6 本章练习 228
 - 12.7 相关资料 228
- 第13章 REST风格的Web Service 229
 - 13.1 何为REST? 230
 - 13.2 REST的6个约束 232
 - 13.2.1 客户端-服务器 233
 - 13.2.2 统一接口 233
 - 13.2.3 无状态 233
 - 13.2.4 可缓存 233
 - 13.2.5 分层系统 234
 - 13.2.6 按需编码 234
 - 13.3 REST API的理查森能力成熟度模型 234
 - 13.3.1 级别0: POX(Plain Old XML)沼泽 235
 - 13.3.2 级别1: 资源 235
 - 13.3.3 级别2: HTTP动词 235
 - 13.3.4 级别3: 超媒体控件 235
 - 13.4 设计REST风格的API 235
 - 13.4.1 资源命名 236
 - 13.4.2 名词而非动词 237
 - 13.4.3 自说明 237
 - 13.4.4 复数而非单数 237
 - 13.4.5 HTTP方法 238
 - 13.4.6 GET 238
 - 13.4.7 POST 239
 - 13.4.8 PUT 239
 - 13.4.9 DELETE 240
 - 13.5 REST实战 240
 - 13.5.1 users名词 240
 - 13.5.2 topics名词与posts名词 242
 - 13.6 使用Java EE实现REST 244
 - 13.7 HATEOAS 249
 - 13.8 何处以及何时该使用REST 253
 - 13.9 小结 253
 - 13.10 本章练习 254
 - 13.11 相关资料 254
- 第14章 模型、视图与控制器模式 255
 - 14.1 何为MVC设计模式? 256

14.2	使用普通代码实现MVC模式	260
14.3	使用Java EE实现MVC模式	265
14.4	FacesServlet	265
14.5	使用FacesServlet实现MVC	266
14.6	何处以及何时该使用MVC模式	269
14.7	小结	269
14.8	本章练习	270
14.9	相关资料	270
第15章	Java EE中的其他模式	271
15.1	何为WebSockets?	272
15.2	何为面向消息的中间件	275
15.3	何为微服务架构?	277
15.3.1	单块架构	278
15.3.2	可伸缩性	279
15.3.3	分解为服务	280
15.3.4	微服务的好处	281
15.3.5	天下没有免费的午餐	282
15.3.6	结论	283
15.4	一些反模式	284
15.4.1	超级类	284
15.4.2	面条架构	284
15.4.3	哥伦布先生	285
15.4.4	利益之交	285
15.4.5	前沿技术	286
15.4.6	辅助类	286
15.5	相关资料	287
第III部分	总结	
第16章	设计模式：好处、坏处与丑陋之处	291
16.1	好处：通往成功之路的模式	292
16.2	坏处：模式的过度使用与滥用	294
16.3	丑陋之处	296
16.4	小结	298
16.5	相关资料	298
• • • • •	(收起)	

[Java EE设计模式解析与应用_下载链接1](#)

标签

设计模式

Java

评论

本书介绍了Java EE相关设计模式。遗憾的是，其中很多例子使用到了EJB来讲解。

书本不厚，一周就可读完，讲的还算详细。不足在于javaee例子为EJB，现阶段的spring还需自己研究应用

[Java EE设计模式解析与应用_下载链接1](#)

书评

[Java EE设计模式解析与应用_下载链接1](#)