

<<Java设计模式>>

图书基本信息

书名：<<Java设计模式>>

13位ISBN编号：9787115156884

10位ISBN编号：7115156883

出版时间：2007-3

出版时间：人民邮电

作者：梅特斯克

页数：309

字数：472000

译者：龚波,赵彩琳,陈蓓

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<Java设计模式>>

内容概要

本书通过最新的Java特征和最佳实践阐释了经典著作《设计模式》中介绍的23种基本设计模式。作者根据自己丰富的编程经验，用实际的Java 5.0程序、清晰的UML图表以及引人入胜的实践讲述了每个模式。

书中每章集中讨论一种特定的模式，之后给出一些自我突破题，自我突破题或者启发读者思考，或者要求编写代码来解决某个问题，有助于更深入地理解书中的概念。

本书适合于各层次的Java开发人员阅读。

<<Java设计模式>>

作者简介

Steven John Metsker是世界知名的面向对象技术专家，在设计模式方面造诣颇深。除本书外，他还出版了Building Parsers with java和Design Patterns in C#等著作，并为许多技术杂志撰写了大量技术文章。

<<Java设计模式>>

书籍目录

| | | | |
|---|-------------------------------------|----------------|------------------------------------|
| 第1章 设计模式介绍 | 11.1 为什么使用模式 | 11.2 为什么使用设计模式 | 21.3 为什么使用Java |
| 21.4 UML | 31.5 自我突破 | 31.6 本书的组织方式 | 31.7 欢迎来到Oozinoz公司 |
| 41.8 小结 | 5 | | |
| 第一部分 接口型模式 | 第2章 接口型模式介绍 | | |
| 82.1 接口和抽象类 | 82.2 接口和责任 | 92.3 小结 | |
| 102.4 超越普通接口 | 10第3章 Adapter (适配器) 模式 | | |
| 113.1 接口适配 | 113.2 类和对象适配器 | | 143.3 为JTable适配数据 |
| 163.4 标识适配器 | 203.5 小结 | | 20第4章 Facade (外观) 模式 |
| 214.1 外观类、工具类和示例类 | 214.2 重构为Facade模式 | | 224.3 小结 |
| 29第5章 Composite (组合) 模式 | | | |
| 305.1 常见的组合 | 305.2 Composite模式的递归特性 | | 315.3 组合、树和环 |
| 325.4 含有环的Composite模式 | 355.5 环的影响 | | 385.6 小结 |
| 39第6章 Bridge (桥接) 模式 | | | |
| 406.1 经典范例：普通抽象 | 406.2 从抽象到Bridge模式 | | 426.3 应用Bridge模式的驱动程序 |
| 436.4 数据库驱动程序 | 446.5 小结 | | 45第二部分 责任型模式 |
| 第7章 责任型模式介绍 | | | |
| 487.1 常见的责任型模式 | 487.2 通过可见性属性控制责任 | | 497.3 小结 |
| 507.4 超越普通责任型模式 | 50第8章 Singleton (单例) 模式 | | |
| 518.1 Singleton模式机制 | 518.2 单例和线程 | | 528.3 识别单例 |
| 538.4 小结 | 54第9章 Observer (观察者) 模式 | | |
| 559.1 经典范例：GUI中的Observer模式 | 559.2 模型/视图/控制器 | | |
| 589.3 维护Observable类对象 | 629.4 小结 | | 64第10章 Mediator (中介者) 模式 |
| 6510.1 经典范例：GUI的Mediator模式 | 6510.2 利用Mediator模式管理关系完整性 | | 6810.3 小结 |
| 73第11章 Proxy (代理) 模式 | | | |
| 7411.1 经典范例：图像代理 | 7411.2 重新思考图像代理 | | 7811.3 远程代理 |
| 7911.4 动态代理 | 8411.5 小结 | | |
| 87第12章 Chain of Responsibility (责任链) 模式 | | | |
| 8812.1 常见的责任链 | 8812.2 重构为Chain of Responsibility模式 | | 8912.3 固定责任链 |
| 912.4 不带组合结构的Chain of Responsibility模式 | 9212.5 小结 | | 93第13章 Flyweight (享元) 模式 |
| 9413.1 不变性 | 9413.2 提取享元中不可变的部分 | | 9513.3 共享享元 |
| 9613.4 小结 | 99第三部分 构造型模式 | | |
| 第14章 构造型模式介绍 | | | |
| 10214.1 普通构造的挑战 | 10214.2 小结 | | 10314.3 超越普通构造 |
| 103第15章 Builder (生成器) 模式 | | | |
| 10515.1 常见的生成器 | 10515.2 根据约束构造对象 | | 10715.3 根据不完整信息构造符合约束的对象 |
| 10915.4 小结 | 109第16章 Factory Method (工厂方法) 模式 | | |
| 11016.1 经典范例：迭代器 | 11016.2 识别Factory Method模式 | | 11116.3 决定要实例化的对象 |
| 11116.4 并行层次结构中的Factory Method模式 | 11216.5 小结 | | 114第17章 Abstract Factory (抽象工厂) 模式 |
| 11517.1 经典范例：GUI工具包 | 11517.2 抽象工厂和工厂方法 | | 11817.3 包和抽象工厂 |
| 12117.4 小结 | 122第18章 Prototype (原型) 模式 | | |
| 12318.1 作为工厂的原型 | 12318.2 利用克隆进行原型化 | | 12418.3 小结 |
| 126第19章 Memento (备忘录) 模式 | | | |
| 12719.1 经典范例：使用备忘录实现撤销操作 | 12719.2 备忘录的持久性 | | 13219.3 跨越会话的持久性备忘录 |
| 13219.4 小结 | 135第四部分 操作型模式 | | |
| 第20章 操作型模式介绍 | | | |
| 13820.1 操作和方法 | 13820.2 签名 | | 13920.3 异常 |
| 13920.4 算法和多态性 | 14020.5 小结 | | 14120.6 超越普通操作 |
| 141第21章 Template Method (模板方法) 模式 | | | |
| 14221.1 经典范例：排序 | 14221.2 完成算法 | | 14521.3 Template Method模式钩子 |
| 14721.4 重构为Template Method模式 | 14821.5 小结 | | 149第22章 State (状态) 模式 |
| 15022.1 状态建模 | 15022.2 重构为State模式 | | 15322.3 使状态成为常量 |
| 15622.4 小结 | 157第23章 Strategy (策略) 模式 | | |
| 15823.1 策略建模 | 15823.2 重构为Strategy模式 | | 16023.3 比较Strategy模式和State模式 |
| 16323.4 比较Strategy模式和Template Method模式 | 16423.5 小结 | | |
| 164第24章 Command (命令) 模式 | | | |
| 16524.1 经典范例：菜单命令 | 16524.2 使用Command模式提供服务 | | 16724.3 Command模式钩子 |
| 16824.4 Command模式与其他模式的关系 | 16924.5 小结 | | 170第25章 Interpreter (解释器) 模式 |
| 17125.1 Interpreter模式范例 | 17125.2 解释器、语言和解析器 | | 18025.3 小结 |
| 180第五部分 扩展型模式 | | | |
| 第26章 扩展型模式介绍 | | | |
| 18426.1 面向对象设计原则 | 18426.2 Liskov替换原则 (LSP) | | |
| 18426.3 Demeter法则 | 18526.4 消除代码坏味 | | 18626.5 超越普通的扩展 |
| 18726.6 小结 | 187第27章 Decorator (装饰器) 模式 | | |
| 18827.1 经典范例：流和输出器 | 18827.2 函数包装器 | | 19327.3 与其他模式相关的Decorator模式 |
| 19927.4 小结 | 199第28章 Iterator (迭代器) 模式 | | |
| 20028.1 常规迭代 | 20028.2 线程安全的迭代 | | 20128.3 对组合结构进行迭代 |
| 20628.3.1 组合枚举器的深度 | 21128.3.2 枚举叶节点 | | 21228.4 小结 |
| 213第29章 Visitor (访问者) 模式 | | | |

<<Java设计模式>>

21429.1 Visitor模式机制 21429.2 常见的Visitor模式 21629.3 Visitor模式循环 22029.4
 Visitor模式危机 22329.5 小结 224第六部分 附录附录A 指南 226A.1 从本书中学到更多
 226A.2 理解设计模式的典型范例 226A.3 在实践中应用设计模式 227A.4 不断地学习
 228附录B 参考答案 229B.1 接口型模式介绍(第2章) 229B.2 Adapter(适配器)模式(第3
 章) 230B.3 Façade(外观)模式(第4章) 233B.4 Composite(组合)模式(第5章)
 235B.5 Bridge(桥接)模式(第6章) 237B.6 责任型模式介绍(第7章) 239B.7 Singleton(单
 例)模式(第8章) 240B.8 Observer(观察者)模式(第9章) 241B.9 Mediator(中介者)
 模式(第10章) 245B.10 Proxy(代理)模式(第11章) 248B.11 Chain of Responsibility(责任
 链)模式(第12章) 249B.12 Flyweight(享元)模式(第13章) 252B.13 构造型模式介绍(第
 14章) 254B.14 Builder(生成器)模式(第15章) 255B.15 Factory Method(工厂方法)模
 式(第16章) 257B.16 Abstract Factory(抽象工厂)模式(第17章) 260B.17 Prototype(原型
)模式(第18章) 262B.18 Memento(备忘录)模式(第19章) 264B.19 操作模式介绍(第20
 章) 266B.20 Template Method(模板方法)模式(第21章) 267B.21 State(状态)模式(第22
 章) 268B.22 Strategy(策略)模式(第23章) 270B.23 Command(命令)模式(第24章)
 271B.24 Interpreter(解释器)模式(第25章) 274B.25 扩展型模式介绍(第26章) 275B.26
 Decorator(装饰器)模式(第27章) 276B.27 Iterator(迭代器)模式(第28章) 278B.28
 Visitor(访问者)模式(第29章) 279附录C Oozinoz源代码 282C.1 获取和使用源代码
 282C.2 构建Oozinoz源代码 282C.3 使用JUnit测试代码 282C.4 自己查找文件 283C.5 小
 结 283附录D UML概览 284D.1 类 284D.2 类间关系 285D.3 接口 286D.4 对象
 287D.5 状态 288词汇表 289参考文献 294索引 296设计模式列表 308

<<Java设计模式>>

媒体关注与评论

最通俗实用的Java模式图书！

<<Java设计模式>>

编辑推荐

《Java设计模式》适合于各层次的Java开发人员阅读。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>