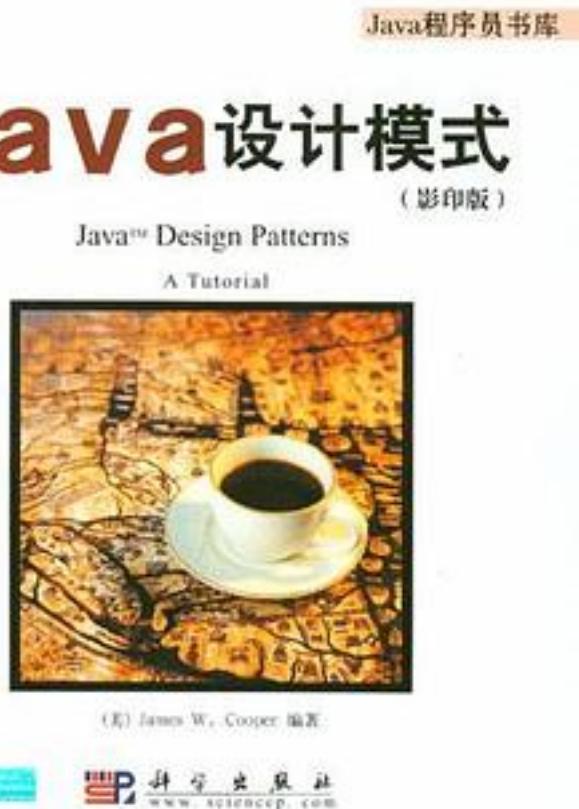


# Java设计模式



[Java设计模式 下载链接1](#)

著者:梅特斯克

出版者:人民邮电出版社

出版时间:2007-3

装帧:

isbn:9787115156884

《Java

《Java设计模式》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。

《Java设计模式》适合于各层次的Java开发人员阅读。

作者介绍：

目录: 第1章 设计模式介绍 1

1.1 为什么使用模式 1

1.2 为什么使用设计模式 2

1.3 为什么使用Java 2

1.4 UML 3

1.5 自我突破 3

1.6 本书的组织方式 3

1.7 欢迎来到Oozinoz公司 4

1.8 小结 5

第一部分 接口型模式

第2章 接口型模式介绍 8

2.1 接口和抽象类 8

2.2 接口和责任 9

2.3 小结 10

2.4 超越普通接口 10

第3章 Adapter (适配器) 模式 11

3.1 接口适配 11

3.2 类和对象适配器 14

3.3 为JTable适配数据 16

3.4 标识适配器 20

3.5 小结 20

第4章 Facade (外观) 模式 21

4.1 外观类、工具类和示例类 21

4.2 重构为Facade模式 22

4.3 小结 29

第5章 Composite (组合) 模式 30

5.1 常见的组合 30

5.2 Composite模式的递归特性 31

5.3 组合、树和环 32

5.4 含有环的Composite模式 35

5.5 环的影响 38

5.6 小结 39

第6章 Bridge (桥接) 模式 40

6.1 经典范例：普通抽象 40

6.2 从抽象到Bridge模式 42

6.3 应用Bridge模式的驱动程序 43

6.4 数据库驱动程序 44

6.5 小结 45

第二部分 责任型模式

第7章 责任型模式介绍 48

7.1 常见的责任型模式 48

7.2 通过可见性属性控制责任 49

- 7.3 小结 50
- 7.4 超越普通责任型模式 50
- 第8章 Singleton (单例) 模式 51
  - 8.1 Singleton模式机制 51
  - 8.2 单例和线程 52
  - 8.3 识别单例 53
  - 8.4 小结 54
- 第9章 Observer (观察者) 模式 55
  - 9.1 经典范例：GUI中的Observer模式 55
  - 9.2 模型/视图/控制器 58
  - 9.3 维护Observable类对象 62
  - 9.4 小结 64
- 第10章 Mediator (中介者) 模式 65
  - 10.1 经典范例：GUI的Mediator模式 65
  - 10.2 利用Mediator模式管理关系完整性 68
  - 10.3 小结 73
- 第11章 Proxy (代理) 模式 74
  - 11.1 经典范例：图像代理 74
  - 11.2 重新思考图像代理 78
  - 11.3 远程代理 79
  - 11.4 动态代理 84
  - 11.5 小结 87
- 第12章 Chain of Responsibility (责任链) 模式 88
  - 12.1 常见的责任链 88
  - 12.2 重构为Chain of Responsibility模式 89
  - 12.3 固定责任链 91
  - 12.4 不带组合结构的Chain of Responsibility模式 92
  - 12.5 小结 93
- 第13章 Flyweight (享元) 模式 94
  - 13.1 不变性 94
  - 13.2 提取享元中不可变的部分 95
  - 13.3 共享享元 96
  - 13.4 小结 99
- 第三部分 构造型模式
- 第14章 构造型模式介绍 102
  - 14.1 普通构造的挑战 102
  - 14.2 小结 103
  - 14.3 超越普通构造 103
- 第15章 Builder (生成器) 模式 105
  - 15.1 常见的生成器 105
  - 15.2 根据约束构造对象 107
  - 15.3 根据不完整信息构造符合约束的对象 109
  - 15.4 小结 109
- 第16章 Factory Method (工厂方法) 模式 110
  - 16.1 经典范例：迭代器 110
  - 16.2 识别Factory Method模式 111
  - 16.3 决定要实例化的对象 111
  - 16.4 并行层次结构中的Factory Method模式 112
  - 16.5 小结 114
- 第17章 Abstract Factory (抽象工厂) 模式 115
  - 17.1 经典范例：GUI工具包 115
  - 17.2 抽象工厂和工厂方法 118
  - 17.3 包和抽象工厂 121
  - 17.4 小结 122

第18章 Prototype (原型) 模式 123

18.1 作为工厂的原型 123

18.2 利用克隆进行原型化 124

18.3 小结 126

第19章 Memento (备忘录) 模式 127

19.1 经典范例：使用备忘录实现撤销操作 127

19.2 备忘录的持久性 132

19.3 跨越会话的持久性备忘录 132

19.4 小结 135

第四部分 操作型模式

第20章 操作型模式介绍 138

20.1 操作和方法 138

20.2 签名 139

20.3 异常 139

20.4 算法和多态性 140

20.5 小结 141

20.6 超越普通操作 141

第21章 Template Method (模板方法) 模式 142

21.1 经典范例：排序 142

21.2 完成算法 145

21.3 Template Method模式钩子 147

21.4 重构为Template Method模式 148

21.5 小结 149

第22章 State (状态) 模式 150

22.1 状态建模 150

22.2 重构为State模式 153

22.3 使状态成为常量 156

22.4 小结 157

第23章 Strategy (策略) 模式 158

23.1 策略建模 158

23.2 重构为Strategy模式 160

23.3 比较Strategy模式和State模式 163

23.4 比较Strategy模式和Template Method模式 164

23.5 小结 164

第24章 Command (命令) 模式 165

24.1 经典范例：菜单命令 165

24.2 使用Command模式提供服务 167

24.3 Command模式钩子 168

24.4 Command模式与其他模式的关系 169

24.5 小结 170

第25章 Interpreter (解释器) 模式 171

25.1 Interpreter模式范例 171

25.2 解释器、语言和解析器 180

25.3 小结 180

第五部分 扩展型模式

第26章 扩展型模式介绍 184

26.1 面向对象设计原则 184

26.2 Liskov替换原则 (LSP) 184

26.3 Demeter法则 185

26.4 消除代码坏味 186

26.5 超越普通的扩展 187

26.6 小结 187

第27章 Decorator (装饰器) 模式 188

27.1 经典范例：流和输出器 188

27.2 函数包装器	193
27.3 与其他模式相关的Decorator模式	199
27.4 小结	199
第28章 Iterator (迭代器) 模式	200
28.1 常规迭代	200
28.2 线程安全的迭代	201
28.3 对组合结构进行迭代	206
28.3.1 组合枚举器的深度	211
28.3.2 枚举叶节点	212
28.4 小结	213
第29章 Visitor (访问者) 模式	214
29.1 Visitor模式机制	214
29.2 常见的Visitor模式	216
29.3 Visitor模式循环	220
29.4 Visitor模式危机	223
29.5 小结	224
第六部分 附录	
附录A 指南	226
A.1 从本书中学到更多	226
A.2 理解设计模式的典型范例	226
A.3 在实践中应用设计模式	227
A.4 不断地学习	228
附录B 参考答案	229
B.1 接口型模式介绍 (第2章)	229
B.2 Adapter (适配器) 模式 (第3章)	230
B.3 Fa?ade (外观) 模式 (第4章)	233
B.4 Composite (组合) 模式 (第5章)	235
B.5 Bridge (桥接) 模式 (第6章)	237
B.6 责任型模式介绍 (第7章)	239
B.7 Singleton (单例) 模式 (第8章)	240
B.8 Observer (观察者) 模式 (第9章)	241
B.9 Mediator (中介者) 模式 (第10章)	245
B.10 Proxy (代理) 模式 (第11章)	248
B.11 Chain of Responsibility (责任链) 模式 (第12章)	249
B.12 Flyweight (享元) 模式 (第13章)	252
B.13 构造型模式介绍 (第14章)	254
B.14 Builder (生成器) 模式 (第15章)	255
B.15 Factory Method (工厂方法) 模式 (第16章)	257
B.16 Abstract Factory (抽象工厂) 模式 (第17章)	260
B.17 Prototype (原型) 模式 (第18章)	262
B.18 Memento (备忘录) 模式 (第19章)	264
B.19 操作模式介绍 (第20章)	266
B.20 Template Method (模板方法) 模式 (第21章)	267
B.21 State (状态) 模式 (第22章)	268
B.22 Strategy (策略) 模式 (第23章)	270
B.23 Command (命令) 模式 (第24章)	271
B.24 Interpreter (解释器) 模式 (第25章)	274
B.25 扩展型模式介绍 (第26章)	275
B.26 Decorator (装饰器) 模式 (第27章)	276
B.27 Iterator (迭代器) 模式 (第28章)	278
B.28 Visitor (访问者) 模式 (第29章)	279
附录C Oozinoz源代码	282
C.1 获取和使用源代码	282
C.2 构建Oozinoz源代码	282

C.3 使用JUnit测试代码 282

C.4 自己查找文件 283

C.5 小结 283

附录D UML概览 284

D.1 类 284

D.2 类间关系 285

D.3 接口 286

D.4 对象 287

D.5 状态 288

词汇表 289

参考文献 294

索引 296

设计模式列表 308

· · · · · (收起)

[Java设计模式 下载链接1](#)

## 标签

设计模式

java

java设计模式

软件工程

计算机

编程

Java

DesignPattern

## 评论

很薄的一本书。思想重于技术。

粗略过了一遍

学习..

感觉书中的例子距离现实太远了，不是很好读

从来没有看过翻译得这么烂的书

这就是三个人翻译地啊...

感觉非常地糟糕，非常后悔在图书馆挑了这本书

"四人帮"<设计模式>的配合读物

实例复杂了

讲的比较易懂，翻译不错

翻的有点烂……

-----  
看到第五章看不下去了，我敢说的是翻译的确实不咋地

-----  
翻译的还是挺烂的...

-----  
翻了一遍，我表示我看不懂。

-----  
内容不够详细，不适合初学者。

-----  
代码示例太简略了，但总结得很好。不适合新手入门

-----  
写的比较不错！

-----  
很棒的书籍，既能学习java知识，也能学习设计模式，对UML图也能够了解一些。

-----  
[Java设计模式 下载链接1](#)

## 书评

《java设计模式第2版》，与之前看过的一些java方面的各种讲设计模式的书籍，相比讲解很详细易懂，最主要是本书用一个java项目贯穿讲解了23种设计模式，这个很难得，以前看过的书籍基本都是每个模式单独拿出了个例子来讲，都是零零碎碎的，没有很好的体现出多个设计模式...

最近工作之余常常翻番这本书，看看能不能重构一下业务上的代码。翻译的。。。。。

应该说还是不错的。书中对设计模式的把握还是不错的。  
不过由于翻译的缘故，有些地方读起来比较费力。

计算机的设计模式经典之作，翻译也很不错，力荐啦，在这里先给翻译者赞个，不然咱们这种英盲就~~~，谢造福！

非常之清晰的讲解，我觉得非常容易理解，还不了解设计模式的快快快，看看这本书！后面我就不多说了，你会明白的！

一本灰常好的技术类的书哦，终于出中文版了，译者翻译的相当好，可见英语水平很好哦，使我受益匪浅，真的是一本值得阅读的好书，顶顶顶顶顶顶！！！！！！！！！！！  
~~~~~非常好...

很好的一本书，译者翻译的也非常不错，清晰明白，一看就懂。自己英文不好，以前很多地方理解的都不到位，现在这些疑惑都解决了，受益匪浅，绝对值得一读！强力推荐给大家！一定会让你有意想不到的收获！

很好的一本书，译者翻译的也非常不错，清晰明白，一看就懂。自己英文不好...

看了很多英文版的设计模式，gof，headfirst等等，都翻译的比较生硬，无法做到信达雅，感觉这本翻译的还比较贴近汉语，适合中国的码农们阅读~

我靠都说了这么多字了怎么还是字数不够，究竟要多少才能够啊，做的太不人性化了，这样会让很多用户觉得反感

好吧，哈哈，牛人的大作，译者不错，赞一个。评论还太短，豆瓣啊。。。。。。。这个要写多长的评论才能发表啊。。。。我在敲一行字还是提示评论太短就不玩了哈，豆瓣这个交互不科学，不科学，不科学！

早就想看这本书了但是我英文不好，终于有翻译版本了，顶一个！很不错的一本书，值得预读，受益匪浅，做java很有帮助，神器呀。好书值得推荐早就想看这本书了

但是我英文不好，终于有翻译版本了，顶一个！很不错的一本书，值得预读，受益匪浅，做java很有帮助，神器呀...

---

是设计模式里面比较好的啦~ 翻译的有的地方生硬了些  
不过也难免，中文和英文毕竟是两种不同的语言...

---

[Java设计模式 下载链接1](#)