《Effective Java 中文版》

图书基本信息

书名：《Effective Java 中文版》

13位ISBN编号：9787111113850

10位ISBN编号：7111113853

出版时间：2003-1

出版社：机械工业出版社

作者：（美）Joshua Bloch

页数：225

译者：潘爱民

《Effective Java 中文版》

内容概要

本书介绍了在Java编程中57条极具实用价值的经验规则，这些经验规则涵盖了大多数开发人员每天所面临的问题的解决方案。通过对Java平台设计专家所使用的技术的全面描述，揭示了应该做什么，不应该做什么才能产生清晰、健壮的高效的代码。
本书中的每条规则都以简短、独立的小文章形式出现，这些小文章包含了详细而精确的建议，以及对语言中许多细微之处的深入分析，并通过例子代码加以进一步说明。贯穿全书的是通用的语言用法的设计模式，以及一些具有启发意义的技巧的技术。

《Effective Java 中文版》

作者简介

Joshua Bloch是Sun公司的高级工程师，也是“ Java平台核心组 ”的设计师。他设计并实现了获奖的Java Collections Framework和java.math软件包，并且对Java平台的其他部分也做出了贡献。Joshua是许多技术文章和论文的作者，他的关于抽象数据对象复制的博士论文获得过“ ACM杰出博士论文奖 ”提名。他拥有哥伦比亚大学的学士学位和卡耐基-梅隆大学的博士学位。
潘爱民 浙江海宁人，现任职于北京大学计算机科学技术研究所，副研究员；研究方向为信息安全（包括网络安全和公钥技术）和软件开发（包括组件技术和模式）；主要著作有《COM原理与应用》等，译著有《Visual C++技术内幕》（第4版）、《COM本质论》和《C++ Primer中文版》等。

《Effective Java 中文版》

书籍目录

《Effective Java 中文版》

编辑推荐

你正在寻找一本间明扼要地阐述Java精髓的书吗？你希望深入地理解Java程序设计语方吗？你希望编写出清晰、正确、健壮和可重用的代码吗？不用再找了，你手上这本书将会使你实现这些愿望，而且还能提供其他许多你意想不到的好处。 ◆“真希望10年前我就能拥有这本书。可能有人会认为我不需要任何关于Java的书籍，但是我确实需要这本书。” ——James Gosling，Java之父，Sun公司副总裁 ◆“一本非常优秀的书，充满了各种关于使用Java程序设计语言和面向对象程序设计的好的建议。” ——Gilad Bracha，Sun公司计算机科学家，《The Java TM Language Specificationy》（Second Edition）的作者之一 “通过这本书，TedNeward将帮助你实现从一个优秀的Java企业应用开发者向一个伟大的开发者的飞跃！” ——John Croupi, Sun著名工程师，《Core J2EEPatterns》作者之一

《Effective Java 中文版》

精彩短评

1、应该再读一遍
2、发人深省
3、java基础
4、一些看起来很忠厚的代码，可能就是让你抓狂了两天的罪魁祸首。Effective Java 给我的方法来预防这类事儿发生。
5、很多疑难点都是在这搞清楚的，也是一部Java面试必备的书
6、为了面试 也是豁出去了 @@
7、Java 经典之作
8、看得第二版, 值得收藏.
9、这版本是第一版的，前面几章关于object的方法的使用令我耳目一新，后面一些章节更倾向于如何去使用（类，接口，变量，函数等），多线程，异常，序列化这三章以前用的少，现在看没有啥共鸣，只能mark以后在看
10、编程进阶专用
11、学习很多java的必须掌握的内容，强烈推荐。
12、有些建议中肯，有些看着疑惑
13、翻译确实一般，译者之前搞的是C++，有些术语翻译得不是很对。这本书指出了Java编程应该留意的地方，以及相应的建议。对进阶来说很不错，即便当时的Java版本只到1.4。至此我的Java学习也告一段落啦，第一次看书有看吐的感觉，应该是短时间接触太多东西还没消化完的缘故吧。我的学习路线是这样：Java大学基础教程->学校课程+国内某本坑爹教材(几乎没提升)->HeadFirst设计模式->Java编程思想->Effective Java，中间夹杂一些实践，这样的学习过程在现在回顾的话，还是觉得蛮适合新手的（学校学习的那部分就砍了吧）。
14、大师的书，不可不读
15、经验之谈
16、爱民翻译的经典之作，强烈推荐。
17、闻山翻译版的，感觉翻译的不好
18、技术类书籍九成九翻译都是渣！
19、提到了少用继承多复合接口，用静态工厂，重载一些基础方法，异常的处理，线程安全，还有一些编程通用的技巧。比较松散，可以多翻翻目录
20、有点老了
21、经典
22、电子版。。受益匪浅

23、虽然jdk1.4，但还是对java优化有了进一步的认识。
java 编程的Best Practice。
24、读这本书应该是快10年前了，现在很多内容也不一定记得了，而且书也不知道被谁拿跑了（或者这本读时是我借小翁的？），补记录一下
25、8万行了谢谢。。熟悉语言+1
26、Java进阶书
27、看完了，没感觉
28、经典无需多言
29、唯一美中不足的就是翻译了，有些地方比较拗口。

内容以编码建议为主，原理性或者知识性的东西比较少，但是对于如何写好Java，帮助还是很大的。
30、当时看的时候有些看不懂, 有空再重读一下吧
31、4年前读过这本书，再次读仍然觉得有收获，这是一本值得多次翻阅的册子
32、还行，但比起C++经验系列，感觉有所不及其地位。java的经典书比C++少
33、看到一半，然后今天同事说项目要重做了..为了跨平台，不用JAVA了...真是忧伤。其实这本书挺

不错的...最后还是看完了，写的太精彩了。
34、中文翻译的有点晦涩
35、去年读的，有些Java的新版本的东西没覆盖到，有很多东西是与面向对象的设计有关，总体来说收获不少。
36、有同类型的C++的书，不同出版社给买了本
因为有个出版社中文译名叫高效java程序设计，内容一样的。
37、比effective c++多得多的条款，感触比较深的是面试官提问说“ effective java里面有一个条款说singleton模式在lazy load和高并发的情况下会出现一些问题，条款里有一种解决方法叫double check，请问什么是double check？后来查了一下无非是饿汉的一些东西，然后程序员的自我修养里面还扯到一些cpu调度顺序，编译器自动优化，然后需要设置barrier才能完美解决、
38、让代码更讲究。
39、必读课本
40、57条极具实用价值的经验规则
41、无论是Java还是.NET，使用带面向对象的，带垃圾收集的语言的程序员们，都应该看看这本书吧。
42、经典~~~
43、有些没有实践过，不是很明白，不过实践过的得到了印证和证明
44、半字典~
45、扎實而穩定的！
46、潘爱民翻译的是比那第一本好多了
47、有助于进一步深刻理解 Java 的书籍
48、经验之谈，值得学习
49、google android官方文档的优化教程中引用了一些Effective Java的原则,其影响力可见一斑。正如Javascript精粹,Effective xx的目标也是鼓励摈弃不良的东西,推荐使用好的实践方案。
50、用到java的时候总可以翻开看看

《Effective Java 中文版》

精彩书评

1、很早就读过，当时就知道这本书很好，可惜当时功力尚浅，没什么收获。但近日再读时，确实很有收获，可以说此书虽不是深入骨髓，但也算入木三分。新手勿动！
2、曾经看过《Effective C++》，《More Effective C++》, 受益匪浅。后来看到《Effective Java》，发现介绍的东西也还不错，就是东西太少，一本书半个小时就看完了，不像前两本书需要不断去感悟
3、内容讲解得很到位。。。。。。。。。。。。。。。。。。。。。。内容讲解得很到位。。。。。。。。。。。。。。。。。。。。。。内容讲解得很到位。。。。。。。。。。。。。。。。。。。。。。内容讲解得很到位。。。。。。。。。。。。。。。。。。。。。。内容讲解得很到位。。。。。。。。。。。。。。。。。。。。。。内容讲解得很到位。。。。。。。。。。。。。。。。。。。。。。内容讲解得很到位。。。。。。。。。。。。。。。。。。。。。。
4、很薄的一本，却很是经典。收获不在于如何高效编写Java程序，而是如何去思考去设计一个类库。那些Java发展的逸事与失误，比其成功更能让人有所得。
5、非常不错的一本书，对想编写健壮程序的人来说非常有用，对equals、hashcode的讲述和异常的处理是本书的精华。但不得不说2002年的书现在来看有些东西确实已经过时了，但瑕不掩瑜，它仍非常具有指导意义
6、《Effective Java中文版(第2版)》http://www.china-pub.com/195040
7、很薄的一本，却很是经典。收获不在于如何高效编写Java程序，而是如何去思考去设计一个类库。那些Java发展的逸事与失误，比其成功更能让人有所得。

《Effective Java 中文版》

章节试读

1、《Effective Java 中文版》的笔记-第1页

Effective Java是我看的第二本学习相关的书，第一本是Thinking in Java。但是发现编程经验太少，有些条目中的场景不能很好的理解，因为没有遇到过，所以反过头来又看了编程思想，才发现编程思想不是一本专属初学者的，甚至可以说不适合初学者。往往学习过一些Java基础的再反过头去看一看编程思想，才能更加深入的理解思想两个字。现在编程思想又大概的过了一遍了，要继续开始看Effective Java，一个月时间，Nix，fighting！！！

2、《Effective Java 中文版》的笔记-第4章.类和接口

12.使类和成员的可访问能量最小化
设计API为最小的可访问性
13.支持非可变性
14.复合优先于继承
15.要么专门为继承而设计,并给出文档说明,要么禁止继承
16.接口优于抽象类
17.接口只是被用于定义类型
18.优先考虑静态成员类
第5章c语言结构的替代
其他省略
21.用类代替enum结构
第6章.方法
23.检查参数的有效性
24.需要时提供保护性拷贝.
117页

3、《Effective Java 中文版》的笔记-第三章对所有对象通用的方法

7.改写equalse方法的时候请遵守约定
需要改写的情况 一个类有自己的“ 逻辑相等 ”，并且超类也没有改写equalse
8.改写equals总是要改写hasecode
9.总是要改写tostring
10.谨慎的改写clone
11.考虑实现Comparable接口

4、《Effective Java 中文版》的笔记-第1页

设计非可变类（对象）
1.只有一个状态：初始构造时的状态，简单
2.所有的操作都是返回一个新构造的对象或者结果，而不去修改类的状态
函数式编程
3.不需要同步
4.可以方便共享同一对象
5.问题：可能会导致大量的对象创建开销

解决方法一：为这种类对象同时提供一个可变的配套的类，在必要的时候使用可变类
解决方法二：将开销较大的计算结果缓存在可变成员中的方式来解决，后续计算可以直接使用之前的缓存结果
6.除非有很好的理由让一个类成为可变类，否则就应该成为非可变类
7.对于某些类，成为不可变的类是不切实际的，比如Thread类，只能尽量保持其不可变性
比如Java中的TimerTask,它的状态空间被有意的设计的非常小。
构造函数应该构造完全初始化的对象，让约束关系尽早建立起来

继承实际上会依赖于基类的实现细节，而不是依赖与其公开API承诺，破坏了封装性，这种关系是比较脆弱的
而组合则依赖与类的公开承诺，不会破坏其封装性
因此，要么专门为继承而设计，并给出文档说明对于可被重写的类的相关实现细节的承诺（好的文档本应该说明这个类做了什么，而非如何做到，这里
由于封装性被破坏，所以违背了这条原则），实际上这种承诺和API一样，是永久的承诺，要么禁止继承

5、《Effective Java 中文版》的笔记-第1页

1.抽象类不利于增加新的接口（需要调整类的层次），抽象类不利于用来定义混合类型，接口允许多重继承
2.每个接口本质上都是定义了一种新的类型
3.二者可以结合，对每个接口都提供一个抽象的骨架实现类，负责所有与接口实现相关的工作。对于实现多个接口的类，可以将操作派发给不同的抽象骨架实现类。

6、《Effective Java 中文版》的笔记-第二章创建和销毁对象

1.静态工厂方法代替构造器
好处. 1.静态工厂方法有名字。
2.不必在每次调用他们的时候都创建一个新的对象
3.可以返回一个元返回类型的子类对象
4.创建参数化类型的时候,他们是代码变得更加简洁
缺点1.如不含有公有的或受保护的构造器.不能被子类化
2.他们和其他静态方法没有区别。
2.遇到多个构造器参数的时候要考虑用构建器
如果类的构造器或静态构造工厂有多个参数,设计这种模式时,builder模式时不错的选择.
3.使用私有构造器或枚举类型强化singleton属性
实现singleton两种方法，都需要私有构造函数，和静态成员。
1.公有静态成员是final

```
public static final Elvis INSTANCE=new Elvis();
private static Elvis getInstance(){

}
```

2.公有静态工厂方法

```
private static final Elvis INSTANCE=new Elvis();
public  static Elvis getInstance(){
return INSTANCE;
}
```

3.编写单个元素枚举类型

4.通过私有构造器强化不可实例化的能力
用于util类，只包含静态方法和静态域，这个类没有必要实例化。对于API可以不实例化这个类。

5.避免创建不必要的对象
当心无意识的自动装箱
6消除过期的对象引用。
无意识的内存泄露
7。避免使用终结函数

《Effective Java 中文版》

版权说明