

<<编写高质量代码>>

图书基本信息

书名：<<编写高质量代码>>

13位ISBN编号：9787111362593

10位ISBN编号：7111362594

出版时间：2011-12-28

出版时间：机械工业出版社华章公司

作者：秦小波

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>

<<编写高质量代码>>

内容概要

在通往“java技术殿堂”的路上，本书将为你指点迷津！

内容全部由java编码的最佳实践组成，从语法、程序设计和架构、工具和框架、编码风格和编程思想等五大方面，对java程序员遇到的各种棘手的疑难问题给出了经验性的解决方案，为java程序员如何编写高质量的java代码提出了151条极为宝贵的建议。

对于每一个问题，不仅以建议的方式从正反两面给出了被实践证明为十分优秀的解决方案和非常糟糕的解决方案，而且还分析了问题产生的根源，犹如醍醐灌顶，让人豁然开朗。

本书一共12章，第1~3章针对java语法本身提出了51条建议，例如覆写变长方法时应该注意哪些事项、final修饰的常量不要在运行期修改、匿名类的构造函数特殊在什么地方等；第4~9章重点针对jdk api的使用提出了80条建议，例如字符串的拼接方法该如何选择、枚举使用时有哪些注意事项、出现nullpointerexception该如何处理、泛型的多重界限该如何使用、多线程编程如何预防死锁，等等；第10~12章针对程序性能、开源的工具和框架、编码风格和编程思想等方面提出了20条建议。

本书针对每个问题所设计的应用场景都非常典型，给出的建议也都与实践紧密结合。

书中的每一条建议都可能在你的下一行代码、下一个应用或下一个项目中崭露头角，建议你将此书搁置在手边，随时查阅，一定能使你的学习和开发工作事半功倍。

<<编写高质量代码>>

作者简介

秦小波，资深软件开发工程师、系统分析师和架构师（获Sun架构师认证），从软件开发工作10余年，实践经验极其丰富。

资深Java技术专家，Java语言、Spring、Struts 2、Hibernate、iBatis、jBPM等Java技术，在企业级Java应用领域积累了大量工程经验，对ESB、BPEL等整合技术也有较深入的认识。

精通设计模式，对设计模式有深刻的认识和独到见解，而且创造性地提出了自己在大量实践中总结出来的新的设计模式。

他撰写的《设计模式之禅》一书凭借优质的内容和良好的可读性广获读者好评，被誉为“设计模式领域的里程碑之作”。

此外，他还是一位优秀的DBA，具有IBM DB2 DBA资格认证，对海量数据处理有深入的研究。

<<编写高质量代码>>

书籍目录

前言

第1章 java开发中通用的方法和准则

- 建议1：不要在常量和变量中出现易混淆的字母
- 建议2：莫让常量蜕变成变量
- 建议3：三元操作符的类型务必一致
- 建议4：避免带有变长参数的方法重载
- 建议5：别让null值和空值威胁到变长方法
- 建议6：覆写变长方法也循规蹈矩
- 建议7：警惕自增的陷阱
- 建议8：不要让旧语法困扰你
- 建议9：少用静态导入
- 建议10：不要在本类中覆盖静态导入的变量和方法
- 建议11：养成良好习惯，显式声明uid
- 建议12：避免用序列化类在构造函数中为不变量赋值
- 建议13：避免为final变量复杂赋值
- 建议14：使用序列化类的私有方法巧妙解决部分属性持久化问题
- 建议15：break万万不可忘
- 建议16：易变业务使用脚本语言编写
- 建议17：慎用动态编译
- 建议18：避免instanceof非预期结果
- 建议19：断言绝对不是鸡肋
- 建议20：不要只替换一个类

第2章 基本类型

- 建议21：用偶判断，不用奇判断
- 建议22：用整数类型处理货币
- 建议23：不要让类型默默转换
- 建议24：边界，边界，还是边界
- 建议25：不要让四舍五入亏了一方
- 建议26：提防包装类型的null值
- 建议27：谨慎包装类型的大小比较
- 建议28：优先使用整型池
- 建议29：优先选择基本类型
- 建议30：不要随便设置随机种子

第3章 类、对象及方法

- 建议31：在接口中不要存在实现代码
- 建议32：静态变量一定要先声明后赋值
- 建议33：不要覆写静态方法
- 建议34：构造函数尽量简化
- 建议35：避免在构造函数中初始化其他类
- 建议36：使用构造代码块精炼程序
- 建议37：构造代码块会想你所想
- 建议38：使用静态内部类提高封装性
- 建议39：使用匿名类的构造函数
- 建议40：匿名类的构造函数很特殊
- 建议41：让多重继承成为现实

<<编写高质量代码>>

- 建议42：让工具类不可实例化
- 建议43：避免对象的浅拷贝
- 建议44：推荐使用序列化实现对象的拷贝
- 建议45：覆写equals方法时不要识别不出自己
- 建议46：equals应该考虑null值情景
- 建议47：在equals中使用getClass进行类型判断
- 建议48：覆写equals方法必须覆写hashCode方法
- 建议49：推荐覆写toString方法
- 建议50：使用package-info类为包服务
- 建议51：不要主动进行垃圾回收

第4章 字符串

- 建议52：推荐使用string直接量赋值
- 建议53：注意方法中传递的参数要求
- 建议54：正确使用string、stringbuffer、stringbuilder
- 建议55：注意字符串的位置
- 建议56：自由选择字符串拼接方法
- 建议57：推荐在复杂字符串操作中使用正则表达式
- 建议58：强烈 建议使用utf编码
- 建议59：对字符串排序持一种宽容的心态

第5章 数组和集合

- 建议60：性能考虑，数组是首选
- 建议61：若有必要，使用变长数组
- 建议62：警惕数组的浅拷贝
- 建议63：在明确的场景下，为集合指定初始容量
- 建议64：多种最值算法，适时选择
- 建议65：避开基本类型数组转换列表陷阱
- 建议66：aslist方法产生的list对象不可更改
- 建议67：不同的列表选择不同的遍历方法
- 建议68：频繁插入和删除时使用linkedlist
- 建议69：列表相等只需关心元素数据
- 建议70：子列表只是原列表的一个视图
- 建议71：推荐使用sublist处理局部列表
- 建议72：生成子列表后不要再操作原列表
- 建议73：使用comparator进行排序
- 建议74：不推荐使用binarysearch对列表进行检索
- 建议75：集合中的元素必须做到compareto和equals同步
- 建议76：集合运算时使用更优雅的方式
- 建议77：使用shuffle打乱列表
- 建议78：减少hashmap中元素的数量
- 建议79：集合中的哈希码不要重复
- 建议80：多线程使用vector或hashtable
- 建议81：非稳定排序推荐使用list
- 建议82：由点及面，一叶知秋—集合大家族

第6章 枚举和注解

- 建议83：推荐使用枚举定义常量
- 建议84：使用构造函数协助描述枚举项
- 建议85：小心switch带来的空值异常

<<编写高质量代码>>

建议86：在switch的default代码块中增加assertionerror错误

建议87：使用valueOf前必须进行校验

建议88：用枚举实现工厂方法模式更简洁

建议89：枚举项的数量限制在64个以内

建议90：小心注解继承

建议91：枚举和注解结合使用威力更大

建议92：注意@Override不同版本的区别

第7章 泛型和反射

建议93：java的泛型是类型擦除的

建议94：不能初始化泛型参数和数组

建议95：强制声明泛型的实际类型

建议96：不同的场景使用不同的泛型通配符

建议97：警惕泛型是不能协变和逆变的

建议98：建议采用的顺序是list[t]、list[?]、list[object]

建议99：严格限定泛型类型采用多重界限

建议100：数组的真实类型必须是泛型类型的子类型

建议101：注意class类的特殊性

建议102：适时选择getdeclared x x x 和get x x x

建议103：反射访问属性或方法时将accessible设置为true

建议104：使用forname动态加载类文件

建议105：动态加载不适合数组

建议106：动态代理可以使代理模式更加灵活

建议107：使用反射增加装饰模式的普适性

建议108：反射让模板方法模式更强大

建议109：不需要太多关注反射效率

第8章 异常

建议110：提倡异常封装

建议111：采用异常链传递异常

建议112：受检异常尽可能转化为非受检异常

建议113：不要在finally块中处理返回值

建议114：不要在构造函数中抛出异常

建议115：使用throwable获得栈信息

建议116：异常只为异常服务

建议117：多使用异常，把性能问题放一边

第9章 多线程和并发

建议118：不推荐覆写start方法

建议119：启动线程前stop方法是不可靠的

建议120：不使用stop方法停止线程

建议121：线程优先级只使用三个等级

建议122：使用线程异常处理器提升系统可靠性

建议123：volatile不能保证数据同步

建议124：异步运算考虑使用callable接口

建议125：优先选择线程池

建议126：适时选择不同的线程池来实现

建议127：lock与synchronized是不一样的

建议128：预防线程死锁

建议129：适当设置阻塞队列长度

<<编写高质量代码>>

建议130：使用countdownlatch协调子线程

建议131：cyclicbarrier让多线程齐步走

第10章 性能和效率

建议132：提升java性能的基本方法

建议133：若非必要，不要克隆对象

建议134：推荐使用“望闻问切”的方式诊断性能

建议135：必须定义性能衡量标准

建议136：枪打出头鸟—解决首要系统性能问题

建议137：调整jvm参数以提升性能

建议138：性能是个大“咕咚”

第11章 开源世界

建议139：大胆采用开源工具

建议140：推荐使用guava扩展工具包

建议141：apache扩展包

建议142：推荐使用joda日期时间扩展包

建议143：可以选择多种collections扩展

第12章 思想为源

建议144：提倡良好的代码风格

建议145：不要完全依靠单元测试来发现问题

建议146：让注释正确、清晰、简洁

建议147：让接口的职责保持单一

建议148：增强类的可替换性

建议149：依赖抽象而不是实现

建议150：抛弃7条不良的编码习惯

建议151：以技术员自律而不是工人

<<编写高质量代码>>

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>