

图书基本信息

书名：<<Effective Java 第二版 中文版/Sun公司核心技术丛书>>

13位ISBN编号：9787111255833

10位ISBN编号：7111255836

出版时间：2009-1

出版时间：机械工业

作者：Joshua Bloch

页数：287

译者：俞黎敏

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>

前言

如果有一个同事这样对你说，“我的配偶今天晚上在家里制造了一顿不同寻常的晚餐，你愿意来参加吗？”(Spouse of me this night today manufactures the unusual meal in a home . You will join?)这时候你脑子里可能会浮现起三件事情：第一，满脑子的疑惑；第二，英语肯定不是这位同事的母语；第三，同事是在邀请你参加他的家庭晚宴。

如果你曾经学习过第二种语言，并且尝试过在课堂之外使用这种语言，你就该知道有三件事情是必须掌握的：这门语言的结构如何(语法)，如何命名你想谈论的事物(词汇)，以及如何以惯用和高效的方式来表达日常的事物(用法)。

在课堂上大多只涉及前面两点，当你使出浑身解数想让对方明白你的意思时，常常会发现当地人对你的表述忍俊不禁。

程序设计语言也是如此。

你需要理解语言的核心：它是面向算法的，还是面向函数的，或者是面向对象的？你需要知道词汇表：标准类库提供了哪些数据结构、操作和功能(Facility)？你还需要熟悉如何用习惯和高效的方式来构建代码。

关于程序设计语言的书籍通常只是涉及前面两点，或者只是蜻蜓点水般地介绍一下用法。

也许是因为前面两点比较容易编写。

语法和词汇是语言本身固有的特性，但是，用法则反映了使用这门语言的群体的特征。

例如，Java程序设计语言是一门支持单继承的面向对象程序设计语言，在每个方法的内部，它也支持命令式的(面向语句的，statement-Oriented)编码风格。

Java类库提供了对图形显示、网络、分布式计算和安全性的支持。

但是，如何把这门语言以最佳的方式运用到实践中呢？还有一点：程序与口语中的句子以及大多数书籍和杂志都不同，它会随着时间的推移而发生变化。

仅仅编写出能够有效地工作并且能够被别人理解的代码往往是不够的，我们还必须把代码组织成易于修改的形式。

针对某个任务可能会有10种不同的编码方法，而在这10种方法中，有7种方法是笨拙的、低效的或者是难以理解的。

而在剩下的3种编码方法中，哪一种会是最接近该任务的下一年度发行版本的代码呢？目前有大量的书籍可以供你学习Java程序设计语言的语法，包括《The Java Programming Language》[Arnold05](作者Arnold、Gosling和Holmes)，以及《The Java Language Specification》[JLS](作者Gosling、Joy和Bracha)。

同样，与Java程序设计语言相关的类库和API的书籍也不少。

本书解决了你的第三种需求：习惯和高效的用法。

作者Joshua Bloch在Sun公司多年来一直从事Java语言的扩展、实现和使用的工作；他还大量地阅读了其他人的代码，包括我的代码。

他在本书中提出了许多很好的建议，他系统地把这些建议组织起来，旨在告诉读者如何更好地构造代码以便它们能工作得更好，也便于其他人能够理解这些代码，便于将来对代码进行修改和改善的时候不至于那么头疼。

甚至，你的程序也会因此而变得更加令人愉悦、更加优美和雅致。

内容概要

本书介绍了在Java编程中78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、健壮和高效的代码。

本书中的每条规则都以简短、独立的小文章形式出现，并通过例子代码加以进一步说明。

本书内容全面，结构清晰，讲解详细。

可作为技术人员的参考用书。

作者简介

Joshua Bloch是Google公司的首席Java架构师。

是Jolt大奖的获得者。

他曾是Sun公司的杰出工程师，和Transarc公司的高级系统设计师。

Bloch曾带领团队设计和实现过无数的Java平台特性，包括JDK 5.0语言增强版和获奖的Java Collections Framework。

他的著作还包括：《Java Puzzler

书籍目录

译者序序前言致谢第1章 引言第2章 创建和销毁对象 第1条：考虑用静态工厂方法代替构造器 第2条：遇到多个构造器参数时要考虑用构建器 第3条：用私有构造器或者枚举类型强化Singleton属性 第4条：通过私有构造器强化不可实例化的能力 第5条：避免创建不必要的对象 第6条：消除过期的对象引用 第7条：避免使用终结函数第3章 对于所有对象都通用的方法 第8条：改写equals时请遵守通用约定 第9条：改写equals时总要改写hashCode 第10条：始终要改写toString 第11条：谨慎地改写clone 第12条：考虑实现Comparable接口第4章 类和接口 第13条：使类和成员的可访问性最小化 第14条：在公有类中使用访问方法而非公有域 第15条：使非可变性最小化 第16条：复合优先于继承 第17条：要么为继承而设计，并提供文档说明，要么就禁止继承 第18条：接口优于抽象类 第19条：接口只用于定义类型 第20条：类层次优于标签类 第21条：用函数对象表示策略 第22条：优先考虑静态成员类第5章 泛型 第23条：请不要在新代码中使用原生态类型 第24条：消除非受检警告 第25条：列表优先于数组 第26条：优先考虑泛型 第27条：优先考虑泛型方法 第28条：利用有限制通配符来提升API的灵活性 第29条：优先考虑类型安全的异构容器第6章 枚举和注解 第30条：用enum代替int常量 第31条：用实例域代替序数 第32条：用EnumSet代替位域 第33条：用EnumMap代替序数索引 第34条：用接口模拟可伸缩的枚举 第35条：注解优先于命名模式 第36条：坚持使用Override注解 第37条：用标记接口定义类型第7章 方法 第38条：检查参数的有效性 第39条：必要时进行保护性拷贝 第40条：谨慎设计方法签名 第41条：慎用重载 第42条：慎用可变参数（varargs） 第43条：返回零长度的数组或者集合，而不是null 第44条：为所有导出的API元素编写文档注释第8章 通用程序设计 第45条：将局部变量的作用域最小化 第46条：for-each循环优先于传统的for循环 第47条：了解和使用类库 第48条：如果需要精确的答案，请避免使用float和double 第49条：原语类型优先于装箱的原语类型 第50条：如果其他类型更适合，则尽量避免使用字符串 第51条：了解字符串连接的性能 第52条：通过接口引用对象 第53条：接口优先于反射机制 第54条：谨慎地使用本地方法 第55条：谨慎地进行优化 第56条：遵守普遍接受的命名惯例第9章 异常 第57条：只针对异常的条件才使用异常 第58条：对可恢复的条件使用受检异常，对编程错误使用运行时异常 第59条：避免不必要地使用受检的异常 第60条：尽量使用标准的异常 第61条：抛出与抽象相对应的异常 第62条：每个方法抛出的所有异常都要有文档 第63条：在细节消息中包含失败 - 捕获信息 第64条：努力使失败保持原子性 第65条：不要忽略异常第10章 并发 第66条：同步访问共享的可变数据 第67条：避免过多同步 第68条：executor和task优先于线程 第69条：并发工具优先于wait和notify 第70条：线程安全性的文档化 第71条：慎用延迟初始化 第72条：不要依赖于线程调度器 第73条：避免使用线程组第11章 序列化 第74条：谨慎地实现Serializable 第75条：考虑使用自定义的序列化形式 第76条：保护性地编写readObject方法 第77条：对于实例控制，枚举类型优先于readResolve 第78条：考虑用序列化代理代替序列化实例

章节摘录

插图：第2章 创建和销毁对象第4条：通过私有构造器强化不可实例化的能力有时候，你可能需要编写只包含静态方法和静态域类。

这些类的名声很不好，因为有些人在面向对象的语言中滥用这样的类来编写过程化的程序。

尽管如此，它们也确实有它们特有的用处。

我们可以利用这种类，以`java.lang.Math`或者`java.util.Arrays`的方式，把基本类型的值或者数组类型上的相关方法组织起来。

我们也可以通过`java.util.Collections`的方式，把实现特定接口的对象上的静态方法（包括工厂方法，见第1条）组织起来。

最后，还可以利用这种类把`final`类上的方法组织起来，以取代扩展该类的做法。

这样的工具类（utility class）不希望被实例化，实例对它没有任何意义。

然而，在缺少显式构造器的情况下，编译器会自动提供一个公有的、无参的缺省构造器（default constructor）。

对于用户而言，这个构造器与其他的构造器没有任何区别。

在已发行的API中常常可以看到一些被无意识地实例化的类。

企图通过将类做成抽象类来强制该类不可被实例化，这是行不通的。

该类可以被子类化，并且该子类也可以被实例化。

这样做甚至会误导用户，以为这种类是专门为了继承而设计的（见第17条）。

然而，有一些简单的习惯用法可以确保类不可被实例化。

由于只有当类不包含显式的构造器时，编译器才会生成缺省的构造器，因此我们只要让这个类包含私有构造器，它就不能被实例化了。

媒体关注与评论

“ 我很希望10年前就拥有这本书。
可能有人认为我不需要任何Java方面的书籍，但是我需要这本书。
” ——Java之父 James Gosling

编辑推荐

《Effective Java中文版(第2版)》内容全面，结构清晰，讲解详细。

可作为技术人员的参考用书。

编码平添乐趣，程序更加完美，高效成为习惯，工作如此轻松。

你是否正在寻找一本能够更加深入地了解Java编程语言的书，以便编写出更清晰、更正确、更健壮且更易于重用的代码。

不用找了！

《Effective Java中文版(第2版)》为我们带来了共78条程序员必备的经验法则，针对你每天都会遇到的编程问题提出了最有效、最实用的解决方案。

这本经典Jolt获奖作品实属众望所归。

作者对新版进行了彻底的更新。

涵盖了自第1版之后所引入的Java E5和Java SE 6的特性，同时开发了新的设计模式和语言习惯用法，介绍了如何充分利用从泛型到枚举、从注解到自动装箱的各种特性。

每一章都包含几个“条目”，以简洁的形式呈现，自成独立的短文，它们提出了具体的建议，对于Java平台精妙之处的独到见解，以及优秀的代码范例。

每个条目的综合描述和解释都阐明了应该怎么做，不应该怎么做，以及为什么。

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>