Alibaba Group 阿里巴巴集团 | 技术丛书

Broadview www.broadview.com.cn

阿里巴巴 Java开发手册

杨冠宝（孤尽） 编著

Effective Coding

中国工信出版集团 電子工業出版社 PUBLISHING HOUSE OF ELECTRONICS INDUSTRY http://www.phei.com.cn

Alibaba Group 阿里巴巴集团 | 技术丛书

阿里巴巴 Java开发手册

杨冠宝（孤尽） 编著

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本手册的愿景是码出高效，码出质量。它结合作者的开发经验和架构历程，提炼阿里巴巴集团技术团队的集体编程经验和软件设计智慧，浓缩成为立体的编程规范和最佳实践。众所周知，现代软件行业的高速发展对开发者的综合素质要求越来越高，因为不仅是编程相关的知识点，其他维度的知识点也会影响软件的最终交付质量，比如，数据库的表结构和索引设计缺陷可能带来软件的架构缺陷或性能风险；单元测试的失位导致集成测试困难；没有鉴权的漏洞代码易被黑客攻击等。所以，本手册以开发者为中心视角，划分为编程规约、异常日志、单元测试、安全规约、MySQL数据库、工程结构、设计规约七个维度，每个条目下有相应的扩展解释和说明，正例和反例，全面、立体、形象地帮助到开发者的成长和团队代码规约文化的形成。

从严格意义上讲，本手册超越了Java语言本身，明确作为一名合格开发者应该具备的基本素质，因此本手册适合计算机相关行业的管理者和研发人员、高等院校的计算机专业师生、求职者等阅读，希望成为大家如良师益友般的工作手册、工具字典和床头书。

图书在版编目（CIP）数据

阿里巴巴 Java 开发手册 / 杨冠宝编著. —北京：电子工业出版社，2018.1
（阿里巴巴集团技术丛书）
ISBN 978-7-121-33231-9

Ⅰ. ①阿… Ⅱ. ①杨… Ⅲ. ①JAVA 语言－程序设计－技术手册 Ⅳ. ①TP312.8-62

中国版本图书馆 CIP 数据核字（2017）第 304858 号

策划编辑：孙学瑛　康　旭
责任编辑：孙学瑛
印　　刷：中国电影出版社印刷厂
装　　订：中国电影出版社印刷厂
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　　邮编：100036
开　　本：787×1092　1/32　印张：3.5　字数：56.8 千字
版　　次：2018 年 1 月第 1 版
印　　次：2018 年 1 月第 2 次印刷
定　　价：35.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

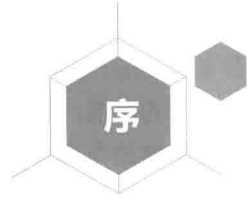
本书咨询联系方式：010-51260888-819，faq@phei.com.cn。

专家语录

“一个优秀的工程师和一个普通工程师的区别，不是满天飞的架构图，他的功底体现在所写的每一行代码上。”

“工程师对于代码，一定要‘精益求精’，不论从性能，还是简洁优雅，都要具备‘精益求精’的工匠精神，认真打磨自己的作品。”

“对程序员来说，关键是骨子里意识到规范也是生产力，个性化尽量表现在代码可维护性和算法效率的提升上。”

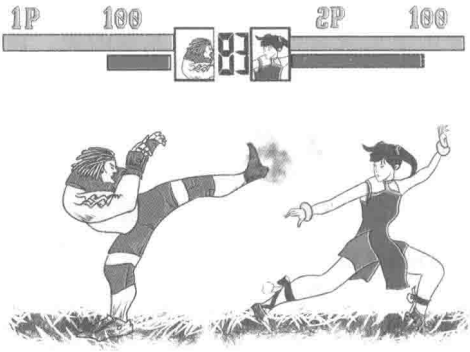


别人都说我们是搬砖的码农，但我们知道自己是追求个性的艺术家。也许我们不会过多在意自己的外表和穿着，但在我们不羁的外表下，骨子里追求着代码的美、系统的美、设计的美，代码规范其实就是一个对程序美的定义。但是这种美离程序员的生活有些遥远，尽管编码规范的价值在业内有着广泛的共识，在现实中却被否定得一塌糊涂。工程师曾经最引以为豪的代码，因为编码规范的缺失、命名的草率而全面地摧毁了彼此的信任，并严重地制约了相互的高效协同。工程师一边吐槽别人的代码，一边写着被吐槽的代码，频繁的系统重构和心惊胆战的维护似乎成了工作的主旋律。

那么如何走出这种怪圈呢？众所周知，互联网公司的优势在于效率，它是企业核心竞争力。体现在产品开发领域，就是沟通效率和研发效率。对于沟通效率的重要性，可以从程序员三大“编程理念之争”说起：

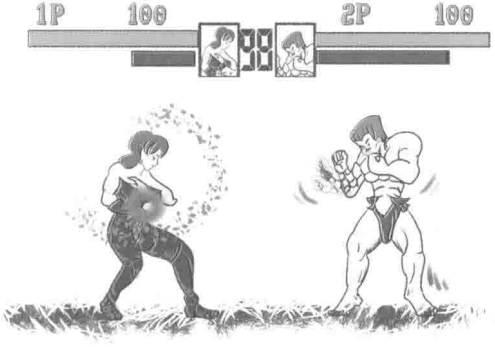
- 缩进采用空格键，还是 Tab 键。
- if 单行语句需要大括号，还是不需要大括号。
- 左大括号不换行，还是单独另起一行。

在美剧《硅谷》中，你也许会记得这样一个经典镜头：主人公 Richard 与同为程序员的女友分手，理由是两人对缩进方式有着不同的习惯，互相拧巴地鄙视着对方的 cody style。Tab 键和空格键的争议在现实编程工作中确实存在。《阿里巴巴 Java 开发手册》（以下简称“《手册》”）明确地支持了 4 个空格的做法，如果一定要问理由，没有理由，因为能够想出来的理由，就像最坚固的盾一样，总有更加锋利的矛会戳破它。只想说，一致性很重要，无边无际争论的时间成本与最后的收益是成反比的。



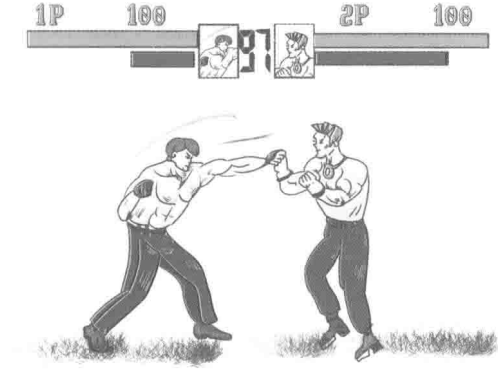
Tab　　空格

if 单语句是否需要换行，也是争论不休的话题。相对来说，写过格式缩进类编程语言的开发者，更加习惯于不加大括号。《手册》中明确 if/for 单行语句必须加大括号，因为单行语句的写法，容易在添加逻辑时引起视觉上的错误判断。此外，if 不加大括号还会有局部变量作用域的问题。



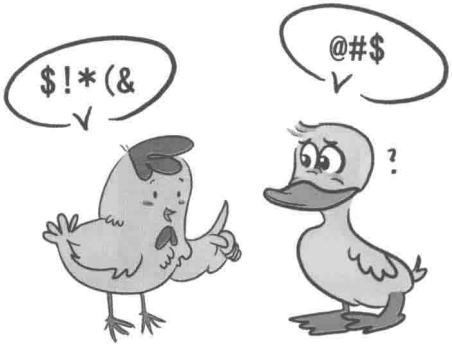
if 单语句不需要大括号　　if 单语句需要大括号

左大括号是否单独另起一行？因为 Go 语言的强制不换行，在这点上，“编程理念之争”的硝烟味没有那么浓。如果一定要给一个理由，那么换行的代码可以增加一行，对于按代码行数考核工作量的公司员工，肯定倾向于左大括号前换行。《手册》明确左大括号不换行。

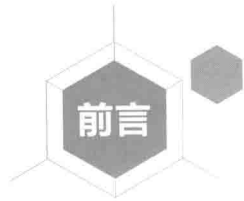


左括号单独一行　　左大括号不换行

这些理念之争的本质就是自己多年代码习惯生的茧，不愿意对不一样的风格妥协，不愿意为了团队的整体效能提升而委屈自己。其实，很多编程方式客观上没有对错之分，一致性很重要，可读性很重要，团队沟通效率很重要。有一个理论叫帕金森琐碎定律：一个组织中的成员往往会把过多的精力花费在一些琐碎的争论上。程序员天生需要团队协作，而协作的正能量要放在问题的有效沟通上。个性化应尽量表现在系统架构和算法效率的提升上，而不是在合作规范上进行纠缠不休的讨论、争论，最后没有结论。规范不一，就像下图中的小鸭子和小鸡对话一样，言语不通，一脸囧相。鸡同鸭讲也恰恰形容了人与人之间沟通的痛点，自说自话，无法达成一致意见。再举个生活中的例子，交通规则靠左行还是靠右行，两者孰好孰坏并不重要，重要的是必须要在统一的方向上通行，表面上限制了自由，但实际上是保障了公众的人身安全。试想，如果没有规定靠右行驶，那样的路况肯定拥堵不堪，险象环生。同样，过分自由随意、天马行空的代码会严重地伤害系统的健康，影响到可扩展性及可维护性。



为了帮助开发人员更好地提高研发效率，阿里巴巴集团基于《手册》内容，独立研发了一套自动化 IDE 检测插件。该插件在扫描代码后，将不符合《手册》的代码按 block/critical/major 三个等级显示在下方；编写代码时，还会给出智能实时提示，告诉你代码如何编写可以更优雅、更符合大家共同的编程风格；对于历史代码，部分规则实现了批量一键修复功能。此插件已经在 2017 年杭州云栖大会上正式对外开放并提供了源码，下载地址：https://github.com/alibaba/p3c。



《阿里巴巴 Java 开发手册》（以下简称“《手册》”）是阿里巴巴集团技术团队的集体智慧结晶和经验总结，经历了多次大规模一线实战的检验及不断的完善，系统化地被整理成册，回馈给广大开发者。

现代软件行业的高速发展对开发者的综合素质要求越来越高，因为不仅是编程知识点，其他维度的知识点也会影响软件的最终交付质量。比如，数据库的表结构和索引设计缺陷可能带来软件的架构缺陷或性能风险，工程结构混乱导致后续维护艰难，没有鉴权的漏洞代码易被黑客攻击等。所以，《手册》以 Java 开发者为中心视角，划分为编程规约、异常日志、单元测试、安全规约、MySQL 数据库、工程结构、设计规约七个维度，再根据内容特征，细分成若干个二级子目录。

根据约束力强弱及故障敏感性，规约依次分为【强制】、【推荐】、【参考】三大类。在规约条目的延伸信息中，“说明”对内容做了适当扩展和解释，“正例”是提倡的编码和实现方式，“反例”是需要提防的雷区及真实的错误案例。

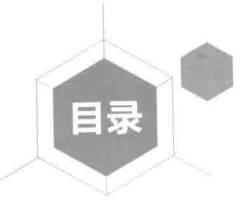
既然《手册》划分为前文所说的七大维度知识点，那么延伸的问题就是“为什么我们会提到这些看似与编码毫无关系的章

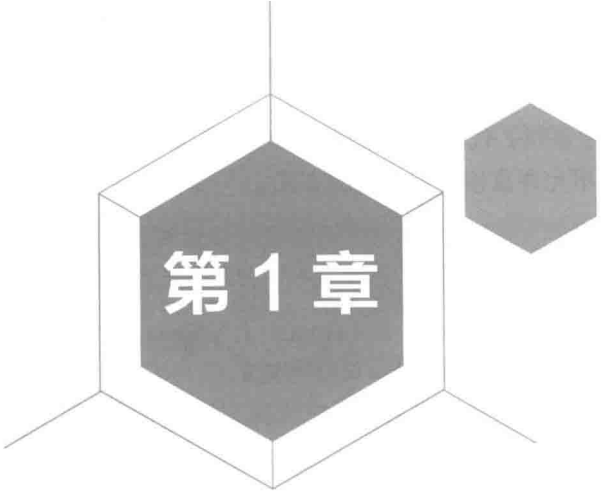
节？”有人曾经质疑，扩展的知识体系本来就是十分庞大的规范文档，比如安全规约扩展开来可以是上百页的资料；与服务器维护相关的 PE 手册厚厚一叠；软件架构设计方向是一个非常复杂的方法论，不知道写进《手册》中的意义何在？其实，《手册》主要关注的是与开发紧密相关的知识点，《手册》不是提倡大家成为安全专家、运维专家，而是关注与编码相关的生态知识，明确了作为一位合格的 Java 开发工程师应具备的基本技术素养。

设计规约部分是本手册独家首发，它根据阿里巴巴一线架构设计经验沉淀而成，旨在帮助研发人员准确度量是否需要定向的设计。近年来，敏捷开发的流行，在一定程度上弱化了设计的重要性，在《手册》中明确了软件设计底线，如果超过规定的阈值，则需要进行有针对性的软件设计与文档沉淀。

最后，《码出高效——阿里巴巴 Java 开发手册详解》即将在 2018 年杭州云栖大会上首发。此书将详细说明规约的初衷和意义、编写和推广历程，每个条目背后的思考与详细的示例代码，以及相应的故障案例分析。拟定的主要章节如下：

第 1 章　从程序员的“编程理念之争”说起
第 2 章　编码规范与团队效能的辩证关系
第 3 章　Java 编程规约剖析
第 4 章　异常日志与问题排查
第 5 章　兢兢业业的单元测试
第 6 章　安全无小事
第 7 章　MySQL 数据库
第 8 章　规范化的工程
第 9 章　设计中体现艺术家的气质





第1章

编程规约

本章是传统意义上的代码规范，包括变量命名、代码风格、控制语句、代码注释等基本的编程习惯，以及从高并发场景中提炼出来的集合处理技巧与并发多线程的注意事项。

1.1 命名风格

1. 【强制】代码中的命名均不能以下画线或美元符号开始，也不能以下画线或美元符号结束。

   反例：_name / __name /$name / name_ / name$ / name__

2. 【强制】代码中的命名严禁使用拼音与英文混合的方式，更不允许直接使用中文的方式。

   说明：正确的英文拼写和语法可以让阅读者易于理解，避免歧义。注意，即使是纯拼音命名方式也要避免采用。

   正例：alibaba / taobao / youku / hangzhou 等国际通用的名称，可视同英文。

   反例：DaZhePromotion [打折] / getPingFenByName() [评分] / int 某变量 = 3

3. 【强制】类名使用 UpperCamelCase 风格，但 DO/BO/ DTO/ VO/AO/PO 等情形例外。

   正例：MarcoPolo / UserDO / XmlService / TcpUdpDeal / TaPromotion / QrCode

   反例：macroPolo / UserDo / XMLService / TCPUDPDeal / TAPromotion / QRCode

4. 【强制】方法名、参数名、成员变量、局部变量都统一使用 lowerCamelCase 风格，必须遵从驼峰形式。

   正例：localValue / getHttpMessage() / inputUserId

5. 【强制】常量命名全部大写，单词间用下画线隔开，力求语义表达完整清楚，不要嫌名字长。

   正例：MAX_STOCK_COUNT/PRIZE_NUMBER_EVERYDAY

   反例：MAX_COUNT/PRIZE_NUMBER

6. 【强制】抽象类命名使用 Abstract 或 Base 开头；异常类命名使用 Exception 结尾；测试类命名以它要测试的类名开始，以 Test 结尾。

7. 【强制】类型与中括号之间无空格相连定义数组。

   正例：定义整形数组 int[] arrayDemo;

   反例：在 main 参数中，使用 String args[]来定义。

8. 【强制】POJO 类中布尔类型的变量都不要加 is 前缀，否则部分框架解析会引起序列化错误。

   反例：定义为基本数据类型 Boolean isDeleted; 的属性，它的方法名称也是 isDeleted()，RPC 框架在反向解析的时候，“误以为”对应的属性名称是 deleted，导致属性获取不到抛出异常。

9 【强制】包名统一使用小写，点分隔符之间有且仅有一个自然语义的英语单词。包名统一使用单数形式，但是类名如果有复数含义，则类名可以使用复数形式。

正例：应用工具类包名为 com.alibaba.ai.util，类名为 MessageUtils（此规则参考 Spring 的框架结构）。

10 【强制】杜绝完全不规范的缩写，避免词不达义。

反例：AbstractClass“缩写”命名成 AbsClass，condition“缩写”命名成 condi，此类随意缩写严重降低了代码的可读性。

11 【推荐】为了达到代码自解释的目标，任何自定义编程元素在命名时，使用尽量完整的单词组合来表达其意。

正例：在 JDK 中，对某个对象引用的 volatile 字段进行原子更新的类名为：AtomicReferenceFieldUpdater。

反例：变量 int a; 的随意命名方式。

12 【推荐】如果模块、接口、类、方法使用了设计模式，应在命名时体现出具体模式。

说明：将设计模式体现在名字中，有利于阅读者快速理解架构设计理念。

正例：
```
public class OrderFactory;
public class LoginProxy;
```

```
public class ResourceObserver;
```

13 【推荐】接口类中的方法和属性不要加任何修饰符号（public 也不要加），保持代码的简洁性，并加上有效的 Javadoc 注释。尽量不要在接口里定义变量，如果一定要定义变量，必须是与接口方法相关的，并且是整个应用的基础常量。

正例：接口方法签名：`void commit();`
接口基础常量：`String COMPANY = "alibaba";`

反例：接口方法定义：`public abstract void commit();`

说明：如果 JDK8 中接口允许有默认实现，那么这个 default 方法，是对所有实现类都有价值的默认实现。

14 接口和实现类的命名有两套规则：

1）【强制】对于 Service 和 DAO 类，基于 SOA 的理念，暴露出来的服务一定是接口，内部的实现类用 Impl 后缀与接口区别。

正例：CacheServiceImpl 实现 CacheService 接口。

2）【推荐】如果是形容能力的接口名称，取对应的形容词为接口名（通常是-able 的形式）。

正例：AbstractTranslator 实现 Translatable。

15 【参考】枚举类名建议带上 Enum 后缀，枚举成员名称需要全大写，单词间用下画线隔开。