Java 编码规范

版本	1.1
作者	李永刚、陈泽洪
发布日期	2014-04-28

目录

1. 目的 4
2. 范围 4
3. 文件组织规范 4
3.1. 编码环境 4
3.2. 文件规则 4
3.3. 文件名规范 4
3.4. 文件组织顺序 4
3.5. 文件注释头规范 5
3.6. 声明规范 5
3.6.1. 包的声明与引用规范 5
3.6.2. 类、接口、变量的声明规范 5
4. 排版规范 6
4.1. 方法分界符 6
4.2. 空格 6
4.3. 缩进 6
4.4. 长度和折行 6
4.5. 方法的规模 6
5. 语句规范 7
5.1. 简单语句 7
5.2. 复合语句 7
5.2.1. 分支语句 7
5.2.2. 循环语句 7
5.2.3. 异常处理语句 7
6. 注释规范 7
7. 命名规范 9
7.1. 包 9
7.2. 类与接口 10
7.3. 方法 10
7.4. 变量 10
7.5. 常量 10
8. 编码规则 10
9. 敏感数据处理规范 11

10. 日志规范 11

11. 附录 12

11.1. 参考资料 12
11.2. 修订记录 13

1. 目的

为统一编码风格，保障程序易维护性和代码安全，并改善可读性，提升升级与修改的效率，特制定本规范。

2. 范围

本规范适用于京东研发部以 Java 为开发语言的软件产品。

3. 文件组织规范

3.1. 编码环境

Java 开发环境编码设置统一采用 UTF-8。

3.2. 文件规则

1）一个程序文件最佳大小在 500 行以内，避免出现超过 1000 行的程序文件;超过 2000 行的程序文件将难以阅读与使用，应尽量避免。

2）一个 Java 源文件应只包含一个单一的公共类或接口。与该公共类关联的私有类和接口可以包含在该公共类和接口的后面。

3.3. 文件名规范

1) 文件名的名字能够标识事物的特性；
2) 文件名的名字一律使用英文单词而不是拼音；
3) 文件名的名字要完整标识，不能使用缩写；
4) 只有大小写不同的文件名会被视为同一个文件，引起文件名冲突。

3.4. 文件组织顺序

文件组织结构应遵循如下顺序：

1. 文件头注释；
2. 包名（package）;

3. 引入（import）声明；
4. 类（class）或者接口（interface）声明。

3.5. 文件注释头规范

所有的源文件在开头需有注释，其中列出文件的版权声明、文件名、功能描述以及创建、修改记录，具体应当遵守以下原则：

1) 必须有版权声明，作者及所属部门的信息；
2) 必须有该类功能及特征描述，要达到使其他人能通过描述明白此类作用的目的；
3) 具体格式如下：
   - 版权信息（@copyright Copyright 2014-XXXX JD.COM All Right Reserved）
   - 与该类相关联类（@see）；
   - 作者（@author XXX 部门：XXXX）；
   - 版本（@version）；
   - 创建、开发日期（@date）；
   - 最后修改日期；
   - 复审人（如有）。

3.6. 声明规范

3.6.1. 包的声明与引用规范

1) 包的引用遵循先核心包后第三方包再其它的顺序原则。

3.6.2. 类、接口、变量的声明规范

1) 每行代码只声明一个变量；
2) 如无特殊需要应尽量在变量声明的同时进行初始化；
3) 类变量声明首先应该声明公共类型的变量（public），再声明保护类型变量（protected），最后声明私有的（private）；
4) 实例变量声明类似于类变量声明，首先应该声明公共类型的（public），再声明保护类型（protected），最后声明私有的类型（private）；
5) 构造器必须作为类的第一个方法声明；
6) 若没有足够的理由，不要把实例或类类变量声明为 public；
7) 必须避免底层声明和高层声明的变量名相同。

4. 排版规范

Java 源文件的编排格式对整体代码的可读性来说意义重大，其目的是为了清晰的可读性和后期的易维护性，应遵循以下规范：

4.1. 方法分界符

每个方法体（包括其注释）应与其它上下方法体（包括其注释）添加一行空行。

4.2. 空格

在两个以上的关键字、变量、常量进行对等操作时，它们之间的操作符之前、之后或者前后要加空格；进行非对等操作时，如果是关系密切的立即操作符（如.），后不应加空格。

4.3. 缩进

程序块要采用缩进风格编写，缩进的空格数为 4 个（缩进不能用 tab），在函数体的开始、类和接口的定义、以及 if、for、do、while、switch、case 语句中的程序或者 static、,synchronized 等语句块中都要采用如上的缩进方式。

4.4. 长度和折行

1) 语句行长度应该控制在 120 个字符以内，超过这个长度应采用折行处理。
2) 折行处理是参照如下规则：
   - 在逗号后折行，
   - 在运算符前折行；
   - 换行后的代码行应该与其同等级的代码行左端对齐。

4.5. 方法的规模

1) 每个方法只实现一个功能；方法的长度（包括方法体内的注释）应尽量控制在 50 行以内，通常不要超过 100 行，不允许出现超过 150 行的情况；
2) 方法的参数应控制在 5 个以内。

5. 语句规范

任何一种程序在编写的过程中，都要用到相应的语句，在 Java 中语句按照自身特点，大致分为简单语句和复合语句。

5.1. 简单语句

1) 每行至多包含一条语句，在同一代码行只能有一个单行计算、赋值语句，不能出现多个；
2) 返回值的 return 语句不使用小括号“()”，除非它们以某种方式使返回值显而易见。
3) 在含有多种运算符的表达式中使用（ ）来界定表达式的组合顺序以避免运算符优先级的问题导致主观意图的错误。

5.2. 复合语句

5.2.1. 分支语句

1) If else 语句后的语句块无论是多条语句还是单条语句都必须通过{}来闭合；
2) Switch 语句最后必须有一条 default case；每个 case 分支结束前必须有 break 语句。

5.2.2. 循环语句

1) 所有的循环语句后面的循环体无论是多条语句还是单条语句都必须使用{}闭合；
2) for 循环语句初始化部分不应该使用超过三个以上的变量；
3) for 循环语句变量的初始化建议尽可能在循环之前进行，除非特殊需要；
4) for/do/while/if/else 等语句块不应该出现无意义空语句块。

5.2.3. 异常处理语句

1) 一般的 Try-catch 结构的语句都必须采用 catch 之后紧跟上一个闭合“}”；
2) 对于某些需保证最终释放资源的操作 Try-catch 必须都加上 finally 分支保证释放处理执行。

6. 注释规范

有效代码注释规模应占总规模的 25%以上。

Java 程序有两类注释：实现注释(implementation comments)和文档注释(document comments)。

- 实现注释使用/*...*/和//界定的注释，用以注释代码或者实现细节；

- 文档注释是 Java 独有的，由/**...*/界定。文档注释用以描述代码的功能规格，可以通过 Javadoc 工具转换成 HTML 文件，即 API 文档。

类别	范例	说明
块注释 (block)	/* * Here is a block comment. */	通常用于对文件，方法数据结构和算法的描述

单行注释
(sigle-line)
	if (condition) {
	/* Handle the condition. */
	...
	}
		短注释可以显示在一行内，并与其后的代码具有一样的缩进层级。

尾端注释
(trailing)
	if (a == 2) {
	return TRUE; /* special case */
	} else {
	return isPrime(a); /* works only for odd a */
	}
		极短的注释可以与他们所要描述的代码位于同一行，但是应该有足够的空白分开代码和注释

行末
(end-of-line)
	if (foo > 1) {
	// Do a double-flip.
	...
	}
		注释定界符//，可以注释掉正行货正一行中的一部分。它一般用于连续多行的注释文本

接口、抽象类、	/**	接口、抽象类、枚举

枚举前注释	* Wms 技术框架学习 Demo * User: YinWei * DateTime: 2011-8-8 11:48:20 * Version 1.0 */ public interface ExampleDao	前注释：必须有创建人、创建时间、用途、版本、以及修改备注
接口中的方法前注释	/** * 根据查询 Bean 获取员工组集合，无翻页 * @param queryBean 员工查询对象 * @return 用户工作组集合 */ public List<BaseWorkerGroup> queryWorkerGroupListByQueryBean(BaseWorkerGroup Query queryBean);	方法的功能描述； 传入参数的说明（@Parma）； 返回参数的说明（@return）；

另外：

1) 开发过程中有未明确逻辑需要保留带后期补充完善的 ,提供 TODO 语法注释。开发完成时将其去掉；
2) 新增代码：写明每个方法的应用场景和参数，对于复杂的算法描述清楚；
3) 维护代码：注释出修改原因、时间、修改人。

7. 命名规范

7.1. 包

1) 包名的前缀总是全部小写的 ASCII 字母并且是一个顶级域名。包名应该独一无二，不可存在重复现象；
2) 格式：com.jd.[系统缩写].[程序].[功能] 按实际情况增加和减少层级。

7.2. 类与接口

1) 类名是一个名词，采用大小写混合的方式，每个单词的首字母大写。尽量使你的类名简洁而富于描述。使用完整单词，避免缩写词(除非该缩写词被更广泛使用，像 URL，HTML)；
2) 接口的大小写规则与类名相似。

7.3. 方法

1) 方法名采用明确具有实际意义的单词,容易通过方法名便知方法的大概用途；
2) 第一个字母小写，在多个单词的情况下第一个单词以后的单词首字母都大写，其余字母小写。

7.4. 变量

1) 除第一个单词的第一个字母小写外，其余单词的第一个字母均为大写，变量名应简短且富于描述；变量名的选用应该易于记忆，即，能够指出其用途；
2) 尽量避免单个字符的变量名，除非是一次性的临时变量。

7.5. 常量

1) 必须采用大写单词命名，多个单词时，单词之间采用下划线加以分割；
2) 对于常量的命名和初始化赋值必须放在类开始部分。

8. 编码规则

1) 未使用的引用和变量必须删掉;
2) 避免用一个对象访问一个类的静态变量和方法,应该用类名替代；
3) 声明常量时要用 final 修饰
4) 在使用局部变量的过程，按就近原则处理。不允许定义一个局部变量，然后在很远的地方才使用；
5) 不要在方法中对方法的参数进行赋值;
6) 避免在程序中使用 System.out 和 System.err 输出信息；
7) 对于对象类型变量比较，不能采用==或!=，而应该使用 equals 方法（null 判断除外）;
8) 泛型必须指定具体类型；
9) 位于 for 循环中作为计数器值的数字常量，除了-1，0 和 1 之外，不应被直接写入代码；
10) 不要将赋值运算符用在容易与相等关系运算符混淆的地方；
11) 内部类规范，在类中定义内部类 InnerClass：
    - 内部类需写在外部类的底部，后面不再有外部类方法；
    - 内部类中的变量定义、函数定义的要求和类中的要求相同；
    - 如果没有设计上的要求，一个内部类应该被声明为 private；一般地，不要把一个内部类声明

为 abstract 或 final；

12) 静态语句类规范中，在类中定义静态语句块:

- 静态语句块的位置应在类静态变量之后，类成员变量之前；
- 在声明类成员变量的时候同时进行初始化。 如非必要，不应把声明时的初始化工作在 static block 中做。

9. 敏感数据处理规范

程序文件中涉及的敏感数据必须按规范处理，敏感数据包括但不限于数据库密码、管理员口令：

1) 对用户可控数据输出必须做转义；
2) 不可直接使用用户提交数据拼接字符串方式查询数据库；
3) 登录页面必须做针对 IP 及用户名登录限制，触犯规则要出现验证码；
4) 用户上传文件后台必须对文件合法性进行检测，存储文件名及路径必须为服务端生成；
5) 文件下载路径变量必须检测，防止越权下载；
6) 不允许在程序中插入类似后门代码；
7) 使用 struts2 框架必须打安全补丁包；
8) 不可明文存储密码，至少经过 3 次 MD5 加密存储。

10. 日志规范

日志输出时，要遵循如下规范：

1) 日志输出前要对 log 级别作判断，不要在代码中固定输出级别；
2) 日志输出时，低级别输出一定要使用 isXXXEnabled 方法；
3) 日志输出级别规范如下：
   (1) trace 任意的跟踪信息，级别最低；
   (2) debug 中间变量内容信息，级别次低；
   (3) info 程序运行时候的正常操作信息，如输入、输出等；
   (4) warning 非正常操作，但不影响最终结果的操作信息；
   (5) error 异常操作，影响结果导致无法继续操作的异常信息；
   (6) fatal 致命异常，导致系统无法正常运行的操作。
4) 为了日后定位问题和业务系统监控，一定使用 log4j 输出日志文件；

5) 日志监控目标：

- 系统接口监控
  - 参数,返回值
  - 调用次数
  - 执行效率
- 业务信息监控：对于复杂算法逻辑将关键数据输出到日志，帮助定位 bug； 比如一段时间内处理了多少订单
- 性能监控
  - 方法执行时间及平均执行时间
  - 方法执行成功次数,失败次数
- 服务器信息
  - 服务器[抓取日志时取服务器 IP]

11. 附录

11.1. 参考资料

序号	规范名称	出处
1.	JAVA Coding Style Guide	SUN
2.	Google Java Style	Google
3.	AMS 开发规范	AMS
4.	FR114-WMS3.0-Java 开发规范 1.1	WMS
5.	Java 代码规范	VCR

6.	Java 编码规范_V1	云平台

11.2. 修订记录

版本	修订内容	作者	发布日期
1.0	文档格式排版，修改目的及注释部分的说明 根据评审意见修改	李永刚、陈泽洪、宋宁	2014-03-03
1.1	增加敏感数据规范	鞠宝松	2014-04-25