**java代码规范**

|  |  |
| --- | --- |
| **类型** | **描述** |
| 缩进 | 使用配置文件进行格式化：  配置文件中一个TAB等于4个空格。 |
| 声明变量、常量 | 一行只声明一个变量或常量；  在代码块的开始处声明实例变量，不要在首次用到该变量时才声明 |
| 语句 | 可以使用eclipse自带的Format方法（快捷键：Ctrl+Shift+F）时 使用eclipse默认的“Control Statements ”格式化方法进行 |
| 包的导入 | 删除不用的导入， 尽量不要使用整个包的导入。 在 eclipse 下经常使用快捷键 ctrl+shift+o 修  正导入。 |
| 域格式 | 每行只能声明一个域。  域的声明用空行隔开 |
| 缩进风格 | 大括号的开始在代码块开始的行尾， 闭合在和代码块同一缩进的行首， 同一层次的代码要保持整齐 |
| 空行的使用 | 空行可以表达代码在语义上的分割， 注释的作用范围，等等。 将类似操作，或一组操作放在  一起不用空行隔开，而用空行隔开不同组的代码 |
| 注释 vs 代码 | 注释宜少二精，不宜多而滥，更不能误导  命名达意，结构清晰， 类和方法等责任明确，往往不需要，或者只需要很少注释，  就可以让人读懂；相反，代码混乱，再多的注释都不能弥补。所以，应当先在代码本  身下功夫。  不能正确表达代码意义的注释，只会损害代码的可读性。  过于详细的注释，对显而易见的代码添加的注释，罗嗦的注释，还不如不写。  注释要和代码同步，过多的注释会成为开发的负担 |
| 块级别注释 | 块级别注释，单行时用 //, 多行时用 /\* .. \*/ 。  较短的代码块用空行表示注释作用域  较长的代码块要用  产品二科/\*------ start: ------\*/  和/\*-------- end: -------\*/  包围 |
| 输出 | 使用 log 而不是 System.out.println()  log 可以设定级别，可以控制输出到哪里，容易区分是在代码的什么地方打印的，而  System.out.print 则不行。 而且， System.out.print 的速度很慢。 所以， 除非是有意的， 否则，  都要用 log。至少在提交到 svn 之前把 System.out.print 换成 log。 |
| 细节 | 每个 if while for 等语句，都不要省略大括号 {} |
| TODO | 在代码中加入 //TODO: ，大部分的 ide 都会帮你提示，让你知道你还有什么事没有做 |
| 程序职责单一 | 关注点分离是软件开发的真理。 人类自所以能够完成复杂的工作， 就是因为人类能够将工作  分解到较小级别的任务上， 在做每个任务时关注更少的东西。 让程序单元的职责单一， 可以  使你在编写这段程序时关注更少的东西，从而降低难度，减少出错。 |
| 缩小变量的作用域 | 能用局部变量的，不要使用实例变量，能用实例变量的， 不要使用类变量。 变量的生存期越  短，以为着它被误用的机会越小， 同一时刻程序员要关注的变量的状态越少。 实例变量和类  变量默认都不是线程安全的，局部变量是线程安全的 |
| 注意 | 尽量不要用参数来带回方法运算结果 |