**Java代码规范**

制作人：马佳慧

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 规范1 | 规范2 | 规范3 |
| 包的命名原则 | 扩展包在“组织名称”后加”x x” | <com>+<组织名称>+<功能>+<具体功能说明> |  |
| 类命名 | <名称>+<功能(类别)> | 前缀单词首字母大写，前缀略写规则见后 | 多个单词紧凑连接，且每个单词首字母大写 |
| 接口命名 | 大小写规则与类名相似 |  |  |
| 变量 | 名称全部由下划线、英文字母和数字组成，变量名不应以下划线或美元符号开头，变量名应能够指出其用途。尽量避免单个字符的变量名，除非是一次性的临时变量。 | 全局变量：一般为多个模块公有， 所以在命名区别于局部变量， 要在一般命名的变量名前加上前缀： g\_；  局部变量：指在一个程序模块中声明使用的变量，该类变量使用 一般变量中定义的命名规则 | 成员变量：是指类成员变量， 用于表示类的一种或一类属性。 要在一般命名的变量名前加上前缀： m\_ |
| 常量 | 类常量和 ANSI 常量的声明，应该全部大写，单词间用下划线隔开。 (尽量避免 ANSI 常量，容易引起错误 ) |  |  |
| Tab 与 Space、Blank Space | 1 Tab = 4 Spaces | 方法与方法间留空行(Blank Space) | 操作符号(=、+、|| 等)两边留一个空格 |
| 大括号{}规则 | 大括号置于关键字或方法、类名下方同列处 |  |  |
| 小括号()及逗号,规则 | <关键字或标识符>+<(左括号>+<参数>+<空格>+<)右括号> | < 标识符 1>+<,逗号>+<空格>+<标识符 2>+… |  |
| 类文件大小 | 每个类文件不应大于 150 行（不包括注释） |  |  |
| 成员方法 | 成员函数的命名应采用完整的英文描述符，大小写混合使用：所有中间单词的第一个字母大写。 | 尽量限制成员函数的可见性。如果成员函数没必要公有 (public)，就定义为保护 (protected)；必要保护 (protected)，就定义为私有 (private)。 |  |
| main()方法 | 公共类(public)必须有 main()方法置于类文件底部，并有测试的简单代码 |  |  |
| 注释书写规范 | Javadoc Tools 注释第一段为本类的用途的简要说明，第二段为概要说明，第三段为算法的概要说明，第四段为作者，按年代顺序排列，第五段为版本号，第六段开发时使用的 JDK 版本 | 对段落(Blocks)与语句的注释，尽量不使用/\*…\*/而使用//注释，并且注释要精简，一般在语  句转折、算法提示、重要成员变量处注释 |  |
| 类结构书写规范 | 资源列表按以下次序排列：  序列一：Java 标准类（Java standard classes）：java.\*.  序列二：Java 扩展类（Java extension classes）：javax.\*.  序列三：第三方厂商类(Third-party classes)  序列四：应用类(Application classes.)  类结构安排：  序列一：Javadoc 注释或者其他头部注释  序列二：类宣告  序列三：成员变量宣告  序列四：构造方法  序列五：方法(按逻辑归类排列)  序列六：内部类  序列七： main()方法（自带一个测试 main()方法） | 方法结构书写规范  规则一：在方法的头部必须有 Javadoc 注释或者类似的注释.  规则二：如果多于 3 个以上的参数，分行书写  规则三：不要在方法名和开括号(“(“)间留空格  规则四：在关括号(“)”)和开大括号间(“{“}保留空格 | 方法或类实例化:  ProductImpl products = new ProductIm (“String 1”,  0,  200,  …);  参数超过 3 个，分行书写 |
| 类、方法、变量命名缩写规范 | 一般方法命名遵循:<动词缩写(首字母小写)>+<名词(首字母大写)>  普通变量遵循:<前缀缩写(首字母小写)>+< 名词(首字母大写)> | final变量遵循:<F>+<”\_ \_” >+< 变量名( 全大写, 各单词间用”\_” 连接)>  static 变量遵循:  <S>+<”\_ \_” >+< 变量名( 首字符大写, 各单词间用”\_” 连接)> | final 且 static 变量遵循:<FS>+<”\_ \_” >+< 变量名( 全大写, 各单词间用”\_” 连接)> |
| 常用 AW T/SW IN G 类命名标准 | Awt/Swing 组件名一般作为定语放在功能前，比如: btnOK 表示为一个确认按钮.如果是 Swing  类命名则在组件缩写后加一个小写的 x，比如 btnxOK 表明是 Swing 组件 Jbutton 按钮。 |  |  |
| 写 bean 类的准则 | 必须有一个不带有参数的公用构造器 | 对于需要的每个特性，应该有一个带有匹配公用 getter 和 setter 方法的专用实例变  量.  对于每个需要的操作，应该有一个公用方法 | 对于需要的每个事件或事件设置，应该定义事件和侦听器类 |
| 语句 | If..else 语句总是用 "{" 和"}" 括起来;  switch每当一个 case顺着往下执行时 (因为没有 break语句)，通常应在 break语句的位置添加注释。每个 switch 语句应包括一个 default选项;  一个 try-catch语句后面也可能跟着一个 finally 语句，不论 try 代码块是否顺利执行完，它都会被执行。 | 复合语句：包含在大括号中的语句序列，形如 "{ 语句 }"  简单语句：每行至多包含一条语句； |  |