

一、文件组织

文件组织规则：

由于超过 2000 行的程序难以阅读，应该尽量避免出现超过 2000 行的程序。一个 Java 源文件都包含一个单一的公共类或接口。若私有类和接口与一个公共类相关联，可以将它们和公共类放入同一个源文件。公共类必须是这个文件中的第一个类或接口。

文件组织顺序：

1. 文件注释：所有的源文件都应该在开头有一个注释，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：

2. 包和引入语句：在多数 Java 源文件中，第一个非注释行是包语句。在它之后可以跟导包语句

3. 类或接口注释：采用 JavaDoc 文档注释，在类、接口定义之前应当对其进行注释，包括类、接口的描述、最新修改者、版本号、参考链接等；

注：JavaDoc 文档注释：描述 Java 的类、接口、构造方法、方法、以及字段。每个文档注释都会被置于注释定界符 `/**...*/` 之中，一个注释对应一个类、接口或成员。该注释应位于声明之前。文档注释的第一行 (`/**`) 不需缩进，随后的文档注释每行都缩进 1 格 (使星号纵向对齐)。

4. 类或接口的实现注释：如果有关类或接口的信息不适合作为“类或接口文档注释”，可以在类或接口的实现注释中给出；

5. 类的（静态）变量：首先是类的公共变量，随后是保护变量，再后是包一级别的变量（没有访问修饰符），最后是私有变量；

6. 实例变量：首先是公共级别的，随后是保护级别的，再后是包一级别的（没有访问修饰符），最后是私有级别的；

7. 构造方法：方法应该按功能分组，而不应该按作用域或访问权限进行分组。

二、代码风格

1. 缩进：程序块要采用缩进风格编写，缩进只使用 TAB 键，不能使用空格键（编辑器中请将 TAB 设置为 4 格）；方法体的开始、类的定义、以及 if、for、do、while、switch、case 语句中的代码都要采用缩进方式；

2. 对齐：程序块的分界符左大括号 "{" 和右大括号 "}" 都另起一行，应各独占一行并且位于同一列，同时与引用它们的语句左对齐；对齐只使用 TAB 键，不使用空格键；不允许把多个短语句写在一行中，即一行只写一条语句；if、for、do、while、case、switch、default 等语句自占一行。

3. 换行：一行的长度超过 80 个字符需要换行，换行规则如下：

在一个逗号后面断开；

在一个操作符前面断开；

长表达式要在低优先级操作符处划分新行；

新行缩进 2 个 TAB。

4. 间隔：类、方法及相对独立的程序块之间、变量说明之后必须加空行；关键字之后要留空格，象 if、for、while 等关键字之后应留一个空格再跟左括号 "("，以突出关键字；方法名与其左括号 "(" 之间不要留空格，以与关键字区别；二元操作符如 "="、"+="、">="、"<="、"+"、"*"、%"、"&&"、"||"、"<<"，"^" 等的前后应当加空格；一元操作符如 "!"、"~"、"++"、"--" 等前后不加空格；xian" []"、"." 这类操作符前后不加空格；for 语句中的表达式应该被空格分开；强制转型后应该跟一个空格。

三、注释

1. 原则：对已经不推荐使用的类和方法需要注明 @Deprecated，并说明替代的类或者方法；对于针对集合、开关的方法，要在方法注释中表明是否多线程安全。

2. 字段注释：采用 JavaDoc 文档注释，定义为 public 的字段必需给出注释，在类的（静态）变量、实例变量定义之前当对其进行注释，给出该字段的描述等

3. 方法注释: 采用 **JavaDoc** 文档注释, 在方法定义之前当对其进行注释, 包括方法的描述、输入、输出及返回值说明、抛出异常说明、参考链接等

4. 单行注释格式//

5. 多行注释格式/*.....*/

四、命名规则

1. 基本规则: 使用可以准确说明变量、字段、类、接口、包等完整的英文描述符; 采用大小写混合, 提高名字的可读性; 采用该领域的术语; 尽量少用缩写, 但如果一定要使用, 当使用公共缩写和习惯缩写等; 避免使用相似或者仅在大小写上有区别的名字。

2. 包命名: 包名一律小写, 少用缩写和长名; 采用以下规则:

[基本包].[项目名].[模块名].[子模块名]...

不得将类直接定义在基本包下, 所有项目中的类、接口等都应当定义在各自的项目和模块包中。

3. 类或接口命名: 类或接口名是个一名词, 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, 避免用缩写词(除非该缩写词被更广泛使用, 像 **URL**, **HTML**)。

4. 变量命名: 采用大小写混合的方式, 第一个单词的首字母小写, 其后单词的首字母大写; 变量名不应以下划线或美元符号开头; 尽量避免单个字符的变量名, 除非是一次性的临时变量。临时变量通常被取名为 **i**, **j**, **k**, **m** 和 **n**, 它们一般用于整型; **c**, **d**, **e**, 它们一般用于字符型; 不采用匈牙利命名法则, 对不易清楚识别出该变量类型的变量应使用类型名或类型名缩写作其后缀; 组件或部件变量使用其类型名或类型名缩写作其后缀; 集合类型变量, 例如数组和矢量, 应采用复数命名或使用表示该集合的名词做后缀。

5. 常量命名: 全部采用大写, 单词间用下划线隔开。

6. 方法命名：方法名是一个动词，采用大小写混合的方式，第一个单词的首字母小写，其后单词的首字母大写；取值类可使用 `get` 前缀，设值类可使用 `set` 前缀，判断类可使用 `is` (`has`) 前缀。

五、声明

1. 类或接口的声名：类、接口定义语法规则如下

```
[ 可见性 ] [ ('abstract' | 'final') ] [ Class | Interface ] class_name
[ ('extends' | 'implements') ] [ 父类或接口名 ] {
    //方法体
}
```

2. 方法声明：良好的程序设计应该尽可能减小类与类之间耦合，所遵循的经验法则是：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`)；没必要保护 (`protected`)，就定义为私有 (`private`)；方法定义语法规则：

```
[可见性] ['abstract'] [ 'static' ] [ 'final' ] [ 'synchronized' ] [返回值类型] method_name(参数列表) [ ('throws') ] [异常列表] {
    //方法体
}
```

声明顺序：构造方法、静态公共方法、静态私有方法、公共方法、友元方法、受保护方法、私有方法、`main` 方法；方法参数建议顺序：(被操作者，操作内容，操作标志，其他)。

3. 变量声明：一行一个声明；声明局部变量的同时初始化（在变量的初始值依赖于某些先前发生的计算的特殊情况下可以不用同时初始化）；只在代码块的开始处声明变量，（一个块是指任何被包含在大括号 "{" 和 "}" 中间的代码）不要在首次用到该变量时才声明；避免声明的局部变量覆盖上一级声明的变量，即不要在内部代码块中声明相同的变量名；公共和保护的可可见性应当尽量避免，所有的字段都建议置为私有，由获取和设置成员函数 (`Getter`、`Setter`) 访问；定义一个变量或者常量的时候，不要包含包名（类似 `java.security.MessageDigest digest = null`），除非是两个包有相同的类名；数组声明时应当将 `[]` 跟在类型后，而不是字段名后；声明顺序：常量、类变量、实例变量、公有字段、受保护字段、友元字段、私有字段。

五、异常

1. 捕捉异常的目的是为了处理它
2. 多个异常应分别捕捉并处理，避免使用一个单一的 `catch` 来处理。

六、习惯

1. `if`、`for`、`do`、`while` 等语句的执行语句部分无论多少都要加括号"`{}`"。
2. 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。
3. 尽量避免在循环中构造和释放对象。
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. 相同的功能不允许复制成 `N` 份代码。
6. 在处理 `String` 的时候要尽量使用 `StringBuffer` 类。