

# **JAVA 语言编程规范**

华云天下©2013

## 目录

1. 介绍/说明.....	4
2. 目标.....	4
3. 命名规定.....	4
3.1 包的命名.....	4
3.2 类的命名.....	4
3.3 接口的命名.....	4
3.4 方法的命名.....	4
3.5 变量的命名.....	5
3.6 常量命名.....	5
3.7 文件的命名.....	5
3.8 推荐的命名.....	5
3.8.1 类名推荐.....	5
3.8.2 Exception 类名推荐.....	5
3.8.3 抽象类名推荐.....	5
3.8.4 Test 类名推荐.....	6
3.8.5 工厂类方法推荐.....	6
4. JAVA 文件组织.....	6
5. JAVA 文件声明顺序.....	6
6. JAVA 文件格式缩进定义.....	7
6.1 缩进尺寸.....	7
6.2 行的尺寸.....	7
6.3 行的格式定义.....	7
7. 注释.....	7
8. 声明.....	7
8.1 变量声明.....	7
8.2 类或接口声明.....	8
8.3 方法声明.....	8
9. 语句格式.....	8
9.1 return 语句.....	8
9.2 if, if-else, if-else-if-else 语句.....	8
9.3 for 语句.....	9
9.4 while 语句.....	9
9.5 do-while 语句.....	9
9.6 switch 语句.....	10
9.7 try-catch 语句.....	10
10. JavaDoc 的格式定义.....	10
10.1 文件头.....	10
10.2 类说明信息.....	11
10.3 变量定义.....	11

10.4 方法定义.....	11
----------------	----

## 1. 介绍/说明

本文档内容描述东华软件 JAVA 编码规范，凡是在东华软件开发的 JAVA 程序必须按照此文档规定。

## 2. 目标

- a. 为来自不同的项目组或个人提供标准的代码格式。
- b. 增加易读性。

## 3. 命名规定

命名规范使程序更易读，从而更易于理解。它们也可以提供一些有关标识符意图的信息，有助于开发人员理解代码。

### 3.1 包的命名

包的命名应该都是小写字母，单词之间用“.”分开。所有的 JAVA 文件必须建立在 com.global 包下。例如：

```
package com.global.water.system;
package com.global.cctv.system;
```

### 3.2 类的命名

类的命名应该都是名词，第一个字母都要大写，其他每个单词的第一个字母都要大写。要用完整的单词，除非是被公认的单词缩写。例如：

```
class Container
class ShippingLine
```

### 3.3 接口的命名

接口的命名应该都是名词或形容词，第一个字母都要大写，其他每个单词第一个字母都要大写。要用完整的单词，除非是被公认的单词缩写。例如：

```
interface ContainerOwner
interface Runnable
```

### 3.4 方法的命名

方法的命名应该都用动词或是惯用短语描述，第一个字母都要小写，其他每个单词第一个字母都要大写。例如：

```
run()
changeLocationTo()
getContainerId()
```

### 3.5 变量的命名

所有非静态变量名的第一个字母都要小写，其他每个单词的第一个字母都要大写。命名应尽量简单并且要有意义。变量名的选用应该易于记忆，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。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。变量名不应以下划线或美元符号开头，尽管这在语法上是允许的。下面是一些正确的变量命名例子：

```
int numOfContainers  
String containerId;  
Date today;
```

### 3.6 常量命名

对于静态的 `final` 变量，在命名的时候每一个单词都要大写，单词之间用 “\_” 分开。例如：

```
final static MIN_WIDTH = 4;  
final static DEFAULT_CONTAINER_SIZE = 20;
```

### 3.7 文件的命名

java 源程序文件以 `.java` 结尾，编译后的文件以 `.class` 结尾。例如：

```
Container.java  
Container.class
```

### 3.8 推荐的命名

#### 3.8.1 类名推荐

当要区别接口和实现类的时候，可以在类的后面加上 “`Impl`”。例如：

```
interface Container  
class ContainerImpl  
class Container3PImpl  
class ContainerYICTImpl
```

#### 3.8.2 Exception 类名推荐

Exception 类最好能用 “Exception” 做为类命名的结尾。例如：

```
DataNotFoundException  
InvalidArgumentException
```

#### 3.8.3 抽象类名推荐

抽象类最好能用 “Abstract” 做为类命名的开头。例如：

```
AbstractBeanDefinition  
AbstractBeanFactory
```

### 3.8.4 Test 类名推荐

Test 类最好能用“Test”做为类命名的结尾。例如：

```
ContainerTest
```

### 3.8.5 工厂类方法推荐

工厂方法最好能把该方法做要创建的对象类型描述出来。例如：

```
public Container createContainer();  
public Location newLocation();
```

## 4. JAVA 文件组织

一个文件由被空行分割而成的段落以及标识每个段落的可选注释共同组成。超过 2000 行的程序难以阅读，所以一个 java 程序文件中的代码行数不能超过 2000 行，除非有特殊原因。

每个 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。

Java 源文件还遵循以下规则，这个规则规定了 java 程序段落的顺序：

- 开头注释
- 包和引入语句
- 类和接口声明

## 5. JAVA 文件声明顺序

类或接口应该按以下顺序声明：

a. 包的定义

b. import 类（输入包的顺序、避免使用\*）

输入包应该按照 java.\*\*, javax.\*\*, org.\*\* , com.\*\*的顺序 import

在 import 的时候不应该使用\*（例如：java.util.\*）

c. 类或接口的定义

d. 静态变量定义，按 public, protected, private 顺序

e. 实例变量定义，按 public, protected, private 顺序

f. 构造方法

g. 方法定义顺序按照 public 方法(类自己的方法)，实现接口的方法，重载的 public 方法，受保护方法，包作用域方法和私有方法。

建议：类中每个方法的代码行数不要超过 100 行。

h. 内部类的定义

## 6. JAVA 文件格式缩进定义

### 6.1 缩进尺寸

用 4 个空格做为缩进尺寸。不建议用 TAB 代替，因为不同的 Text Edit 工具对 TAB 的设置是不同的。

### 6.2 行的尺寸

每行不要超过 80 个字符。

### 6.3 行的格式定义

当一行表达式不能在一行内显示，请按下列顺序要求拆行：

- a. 在 “(” 或 “=” 符号后拆行
- b. 在 “,” 拆行
- c. 在一个操作符后拆行
- d. 把并发的拆行放到同一级别上的缩进
- e. 如果在拆行中再次拆分的时候遇到 “(”，应该新拆出来的行放在更远的一个缩进级别上

```
例如：methodWithLongName(  
    expression1, expression2, expression3,  
    expression4, expression5);  
var =  
method1(  
    expression1, expression2,  
    method2(  
        expression3, expression4));
```

## 7. 注释

Java 有两种注释方法。“/\* This is a comment \*/” 或 “// This is a comment”

第一种应该被用到写 JavaDoc 上，并且都用 “/\*\*” 开头。

第二种适合于在做部分代码的注释，但只适合做非常短内容的注释。

## 8. 声明

### 8.1 变量声明

推荐每行声明一个变量，并加注释。例如：

```
int count; // number of containers  
int size; // size of table
```

```
int count, size;    // AVOID THIS!
```

数组声明应该采用前缀方式。例如：

```
int[] table;
```

```
String[] args;
```

## 8.2 类或接口声明

a. “{” 和声明语句在同一行。

b. 如果不在同一行显示，就将 “extends” 或 “implements” 进行拆行，并放在两个缩进级别后。

c. “}” 符号应该独自占一行。

例如：

```
public class Manager extends Employee {  
  
    ...  
  
}  
  
public class ChiefExecutiveOfficer  
    extends Manager  
    implements Person {  
  
    ...  
  
}
```

## 8.3 方法声明

a. “{” 和声明语句在同一行。

b. “}” 符号应该独自占一行。

例如：

```
public int myMethod(int i, int j) {  
  
    ...  
  
}
```

## 9. 语句格式

### 9.1 return 语句

return 后面的 value 在比较明显的时候不要用 “()”。例如：

```
return;
```

```
return myDisk.size();
```

```
return (size ? size : defaultSize);
```

### 9.2 if, if-else, if-else-if-else 语句

例如：



```
if (condition) {
    statements;
}
if (condition) {
    statements;
}
else {
    statements;
}
if (condition) {
    statements;
}
else if (condition) {
    statements;
}
else if (condition) {
    statements;
}
```

### 9.3 for 语句

例如:

```
for (initialization; condition; update) {
    statements;
}
```

### 9.4 while 语句

例如:

```
while (condition) {
    statements;
}
```

### 9.5 do-while 语句

例如:

```
do {
    statements;
}
while (condition);
```

## 9.6 switch 语句

例如:

```
switch (condition) {  
    case ABC:  
        statements;  
    case DEF:  
        statements;  
        break;  
    case XYZ:  
        statements;  
        break;  
    default:  
        statements;  
        break;  
}
```

## 9.7 try-catch 语句

例如:

```
try {  
    statements;  
}  
catch (ExceptionClass e) {  
    statements;  
}  
finally {  
    statements;  
}
```

## 10. JavaDoc 的格式定义

### 10.1 文件头

应该包括 Copyright, 文件版本等信息。例如:

```
/*  
 * Copyright (C) 2004 北京东华软件有限公司.  
 *  
 * 本系统是商用软件, 未经授权擅自复制或传播本程序的部分或全部将是非法的.
```

```
*  
*  $$Id:$$  
* Date   AuthorDescription  
*/
```

## 10.2 类说明信息

定义文件描述，作者，版本。例如：

```
/**  
 * Class description goes here.  
 *  
 *  
 * @author  <a href="weib@global-tec.net">xxx</a>  
 * @version CVS $$Revision$$ $$Date$$  
*/
```

## 10.3 变量定义

定义变量描述。例如：

```
/**  
 * Comment for <code>${field}</code>  
*/
```

## 10.4 方法定义

定义方法的描述，参数，返回值，参照文档，Exception。例如：

```
/**  
 * description  
 *  
 * @param  
 * @return  
 * @exception  
 * @see  
 * @since  
*/
```