|  |
| --- |
| **《软件编码规范说明书》** |
| 家教管理系统编码规范 |
|  |
|  |
|  |
| **Dare To Dream** |
|  |
|  |

目录

[1.概述 3](#_Toc517194385)

[1.1 编写目的 3](#_Toc517194386)

[1.2 预期读者 3](#_Toc517194387)

[1.3适用范围 3](#_Toc517194388)

[2.命名规范 3](#_Toc517194389)

[2.1包命名规则 3](#_Toc517194390)

[2.1.1与业务系统相关的包命名 3](#_Toc517194391)

[2.1.2与业务系统无关的、可公用的包 3](#_Toc517194392)

[2.1.3类名、接口命名规则 4](#_Toc517194393)

[2.1.4方法命名规则 4](#_Toc517194394)

[2.1.5变量命名规则 4](#_Toc517194395)

[2.1.6常量命名规则 5](#_Toc517194396)

[3.代码风格规范 5](#_Toc517194397)

[3.1返回值 5](#_Toc517194398)

[3.2异常 5](#_Toc517194399)

[3.3表达式 6](#_Toc517194400)

[3.4体前代码 6](#_Toc517194401)

[3.5注释 6](#_Toc517194402)

# 1.概述

## 1.1 编写目的

本文描述了JAVA开发中的有关包、类、接口、方法、实例变量、变量和常量的命名规则，用于规范JAVA编程过程中的命名和代码书写规范。

## 1.2 预期读者

开发组全体人员

## 1.3适用范围

适用于开发组基于JAVA开发的家教服务管理系统项目。

# 2.命名规范

## 2.1包命名规则

开发组将基于JAVA开发中产生的包分为两类，一是与各业务系统相关的包，一是与业务系统无关的、可公用的包。它们的命名规则除了要遵守“包名应全部是小写字母，包名中不能出现下划线，并且第一个字母不能是数字”的规则。

2.1.1与业务系统相关的包命名

与业务系统相关的包命名格式为：

com.wisoft.<projectname>.<modulename>。

其中： <projectname>为项目英文简称或缩写；<modulename>为模块英文名称或简称，如果无细分模块的话可省略模块名。

2.1.2与业务系统无关的、可公用的包

通用包命名格式为：

com. wisoft.common.<modulename>//所有项目通用

com.wisoft.<projectname>.common//单个项目内各模块通用

2.1.3类名、接口命名规则

类和接口的名称应是一个名词，采用大小写混和的方式，如果使用拼音则应是全拼，所有单词都应紧靠在一起，其中每个单词的首字母应大写，接口名称应以大写I开始。例如：

class User；

interface ICommonConnection；

class BuMen;//此处为部门全拼

注意：每个类定义要前必须加类的说明。

2.1.4方法命名规则

方法名应是一个动词或动名结构，采用大小写混和的方式，其中第一个单词的首字母用小写，其后单词的首字母大写。例如：

Xxxxx.getBmList()；

每个方法前必须加说明包括：参数说明、返回值说明、异常说明。如果方法名实在是太长可以对变量名缩写，但是必须添加相应的说明。

2.1.5变量命名规则

变量命名一般采用大小写混和的方式，第一个单词的首字母小写，其后单词的首字母大写，变量名一般不要用下划线或美元符号开头。变量名应简短且有意义，即，能够指出其用途。除非是一次性的临时变量，应尽量避免单个字符的变量名。

（1）类的实例对象定义如下：

Student student;

（2）同一个类的多个对象可以采用一下定义方式：

Student student 1；

Student student 2；

（3）集合类的实例命名使用集合包含元素的英文名称的复数表示，例如：

Vector students;

（4）如果变量名实在是太长可以对变量名缩写，但是必须在类说明或方法说明部分（视缩写的范围而定）进行说明。

（5）数组的声明要用"int[] packets"的形式，而不要用"int packets[]"。

2.1.6常量命名规则

类常量和ANSI常量的命名应全部用大写，单词间用下划线隔开。例如：

final static int MIN\_WIDTH = 4；

final static int MAX\_WIDTH = 99；

# 3.代码风格规范

类的方法的代码行数不能过长，尽量控制在100行（90%），长的方法要拆分成私有函数。

## 3.1返回值

在一般情况下，方法返回值不应返回null。而是尽量使用异常代替返回null。如果在特殊情况必须返回null, 必须在方法说明中加以特别说明，如使用“特别注意”等字样。例如：从一个集合类实例中提取一个对象，因为有些集合类实例是允许null作为键或值的，这个时候用异常取代返回null就不合适了。

如果方法的返回值是集合类对象，而且返回的集合对象不包含任何元素时，则应返回0长度或0大小的集合对象。不能返回null。

## 3.2异常

整个应用系统使用自行设计的唯一异常类，该类包括message（表示错误信息）和ID号（整型，表示异常类型）两部分，该类在创建时是自动获得类名、方法名、行号等信息。

在系统开发和上线之后的一段时间内，异常信息要直接发送到浏览器页面，以便于开发人员迅速定位错误。

## 3.3表达式

（1）所有的算术、逻辑表达式的每一项运算都需要加圆括号，避免使用java语言的运算符优先级，例如：

(2 \*(x + y))/(1 - x)；

((n > 1)?(n - 1):(n = 1))

result =（result && (lastOperand > nextOperand)）；

（2）二元算术运算符（除去“/”）、二元逻辑元素符、赋值运算符，既“+、-、\*、%、+=、-=、\*=、/=、%=、>、<、 ==、 >=、<=、 =”等符号左右两边要加空格，例如：

if(lastOperand >= lastOperand)

（3）参数说明部分的逗号“,”和for语句循环说明部分的分号“;”之前不需要留空格，之后需要留空格。如：

Calculator.add(int a, int b);

for(int i = 0; i < 100; i ++);

## 3.4体前代码

体前代码包括：

a.方法的参数说明和异常说明；

b.条件语句，如if语句、switch语句；

c.循环语句，如while语句、for语句。

这些语句的参数说明、条件说明和循环控制都放在圆括号内。如果不是特别长，应尽量放在同一行内。

同时注意，参数说明、条件说明和循环控制的结束圆括号“)”与体开始花括号“{”之间留一个空格。

## 3.5注释

注释是软件可读性的具体体现。程序注释量一般占程序编码量的20%，软件工程要求不少于20%。以下是四种必要的注释：

类说明注释

注释一般位于 package/import 语句之前，class 描述之前。要求至少写出内容说明、创建者、创建时间和特别注意事项等内容。例如：

/\*\*

\* 名称: ${file\_name}<br>

\* 描述: <br>

\* 类型: JAVA<br>

\* 最近修改时间:${date} ${time}<br>

\* @since ${date}

\* @author MuMay

\*/

（2）方法说明注释

对几乎每个方法都应有适当的说明，位于方法声明之前，包括：说明，参数说明、异常说明、返回值说明和特别说明等。例如：

/\*\*

\* 方法描述

\*

\* ${tags}

\* @param id String唯一标识

\* @param personid String 用户唯一标识

\* @return rtobj BaseReturn 基本返回对象

\* @变更记录 ${date} ${time} MuMay 创建

\*

\*/

（3）体内代码的注释

体(方法体、代码块体、静态代码块体等)内的代码按照功能分成多个虚拟的功能块，每个块以块注释“/\* xxx \*/”注释开始，以空行结束；例如：

/\*\*是否管理员\*\*/

private Boolean isadmin = false;

if(null!=curOuId&&!curOuId. equals (“”))

{

/\*组织机构ID不为空时 \*/

curOuId = ‘000000’;

空行

}

空行

if(curOuId==null)

{

/\*组织机构ID为空时 \*/

}

（4）行注释

行注释“//”仅用于调试注释，在程序稳定之后，行注释必须被删除，以免影响程序的可读性和可移植性。