# 概述

此文档描述的java开发时候编码规范，为了统一一个组的代码规范，让代码看起来是一个人写的。让组内成员在观看别人代码的时候能够迅速了解代码。可能和网上的会有少部分不同【反正大家的编码规范都不一样啦~】我会不断的补充此文档。

# 编码规范

## 包

包命名统一用小写，如 ioinput。如果包名太长可以简写，如businesslogic.userbl

## 类接口枚举

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。【大驼峰命名法】

每个类定义要前必须加类的说明。包括创建时间、最近修改时间、作者、类的描述，使用文档注释【/\*\* \*/】

**例:**

/\*\*

\* 所有po类的父类，包括时间，编号

\* @author cylong

\* @version Oct 25, 2014 11:02:36 PM

\*/

public class PersistentObject

注：上面的author和verson是可以自动生成的，在eclipse的配置里，不会的话问我

枚举规范，每一个枚举必须有一个对应的值，比如：

/\*\*

\*用户的身份：库存管理人员、进货销售人员、财务人员、总经理、销售经理、管理员

\* @author cylong

\* @version Oct 26, 2014 1:25:15 AM

\*/

public enum UserIdentity {

GENERAL\_MANAGER("总经理"),

INVENTORY\_MANAGER("库存管理人员"),

FINANCE\_MANAGER("财务人员"),

PUR\_SALE\_MANAGER("进货销售人员"),

SALE\_MANAGER("销售经理"),

ADMIN("管理员");

public final String value;

UserIdentity(String value) {

this.value = value;

}

}

## 方法

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。【小驼峰命名法】

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，最近修改时间。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。使用文档注释【/\*\* \*/】

**例:**

/\*\*

\* 自动生成编号

\* 由不同的单子决定如何生成编号和编号的格式

\* @return 单据的编号，默认为空

\* @author cylong

\* @version Oct 25, 2014 11:38:18 PM

\*/

protected String createID() {}

## 变量

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，用文档注释说明成员变量变量的应用，局部变量无所谓

/\*\* 变量的说明 \*/

private int a = 0;

另外：

1. 同一个类的多个对象可以采用一下定义方式

Person person1;

Person person2; // 一般不需要

1. 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

Vector persons;

## 常量

常量的命名应全部用大写，单词间用下划线隔，例如

final static int MIN\_WIDTH = 4；

## 表达式

1. 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：
   1. a += b;
   2. a > b;
2. 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：
   1. Calculator.add(int a, int b);
   2. for(int i = 0; i < 100; i ++);

## 前体代码

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。

{} 中的语句应该单独作为一行. 例如, 下面的第1行是错误的, 第2行是正确的:

if (i>0) { i ++ }; // 错误, { 和 } 在同一行

 if (i>0) {

 i ++

  }; // 正确, { 单独作为一行   } 语句永远单独作为一行.

  如果 } 语句应该缩进到与其相对应的 { 那一行相对齐的位置

# 其他

1. 未完成的部分要用 // TODO 标记
2. 文件统一用utf-8编码
3. 不要有任何的黄线警告
4. 如果对别人的代码有所修改请说明修改原因和修改人员，最好保留原来的代码(在修改的代码前面写上注释)
5. 注意写代码的时候注意自己的代码易于修改并且适应性强(比如修改组件的位置，防御式编程)
6. 序列化文件统一用.ser为后缀，数据统一保存在项目目录下的data文件夹下