**编码规范~中软**

**1**.程序块要采用缩进风格编写，缩进的空格数为**4**个

**2**.分界符（如大括号‘{’和‘}’）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。 在函数体的开始、类和接口的定义、以及if、for、do、while、switch、case语句中的程序都要采用 如上的**缩进**方式

**3**.较长的语句、表达式或参数（>**80**字符）要分成多行书写，长表达式要在**低优先级**操作符处划分 新行，操作符放在**新行之首**，划分出的新行要进行适当的**缩进**，使排版整齐，语句可读。

**4**.不允许把多个短语句写在一行中，即**一行只写一条**语句

**5**.if, for, do, while, case, switch, default 等语句**独占一行**，且if, for, do, while等语 句的执行语句无论多少都要**加括号{}**

**6**.相对独立的程序块之间、变量说明之后必须**加空行**

**7.**比较操作符, 赋值操作符"="、 "+="，算术操作符"+"、"%"，逻辑操作符"&&"、"&"，位域 操作符"<<"、"^"等双目操作符的前后**加空格**

**8.** "!"、"~"、"++"、"--"、"&"、"."等单目操作符前后**不加空格**

**9**. 一般情况下，源程序有效注释量必须在**30％**以上

**10**.包的注释写入一个名为 **package.html**的HTML格式的说明文件放入当前路径，方便JavaDoc 收集，简述本包的作用、描述本包的内容、产品模块名称和版本、公司版权 内容：工单公共应用原子服务接口实现  
产品模块名称：BUS  
版本：1.1  
公司版权：Huawei

**11.**文件注释写入**文件头部**，包名之前的位置

/\*\* \* 文件名称：WSForCBSChangeOwnershipPS.java

\* 版本：1.3

\* 描述：修改客户信息的业务逻辑处理

 \* 创建人：czj \* 创建时间：2010-09-19

\* 修改人：czj \* 修改时间：2010-09-25

\* 修改人：wade \* 修改时间：2010-11-25 \*/

**12.项目→模块→包→文件→类（接口）→属性（方法）**

**13**.类和接口的注释放在 package 关键字之后，class 或者 interface 关键字之**前**，内容为类的 注释主要是一句话功能简述、功能详细描述

**14**.类属性、公有和保护方法注释：写在类属性、公有和保护方法**上面**。成员变量注释内容：成员 变量的意义、目的、功能，可能被用到的地方

**15**.注释与所描述内容进行**同样的缩排，**将注释与其上面的代码用**空行隔开**

**16**.遇到数据库操作，IO流操作，解析文件需要try catch finally，在finally中关闭流。

**17**.**边写代码边注释**，修改代码同时修改相应的注释，以保证注释与代码的一致性。不再有用的 注释要删除

**18**.注释的内容要清楚、明了，含义准确，**防止**注释**二义性**

**19**.注释语言不统一，影响程序易读性和外观排版，出于对维护人员的考虑，建议使用**中文**

**20**.包名采用**小写**字母 部门内部产品：部门的名称 + 项目名称 产品线的产品：产品的名称 + 模块的名称

**21**.类名和接口、方法、属性名使用类意义**完整的英文**描述，每个英文单词的首字母使用**大写**、其余 字母使用小写的大小写**混合**法 get + 非布尔属性名() is + 布尔属性名() set + 属性名() 动词() 动词 + 宾语()

**22**．属性名**不能**与方法名相同

**23**.常量名使用全**大写**的英文描述，英文单词之间用**下划线**分隔开，并且使用 final static 修饰

**24**.如果函数名超过**15**个字母，可采用以去掉元音字母的方法或者以行业内约定俗成的缩写方式 缩写函数名。如Information为Info

**25**.准确地确定成员函数的**存取控制**符号，不是必须使用 public 属性的，请使用 protected， 不是必须使用 protected, 请使用 private

**26**.方法内的单行注释使用 **//**

**27**.类属性和类方法不要交叉放置，不同存取范围的属性或者方法也尽量不要交叉放置。**前面一** **般写属性，后面写方法**，范围**从大到小**

**28**.明确方法功能，精确（而不是近似）地实现方法设计。一个函数仅完成一件功能，**即使简** **单功能也应该编写方法实现**

**29**.应**明确规定**对接口方法参数的合法性检查应由方法的调用者负责还是由接口方法本身负责， 缺省是由方法调用者负责

**30**.明确类的功能，精确（而不是近似）地实现类的设计。一个类仅实现一组**相近**的功能。说明： 划分类的时候，应该尽量把逻辑处理、数据和显示分离，实现类功能的**单一性**

**31**.数据库操作、IO操作等需要使用结束close()的对象必须在try -catch-finally 的**finally** 中close()，无论有没有异常，流必须要关闭

**32**.异常捕获后，如果不对该异常进行处理，则应该纪录日志或者ex.printStackTrace()，**不能** **什么都不做**

**33**.在程序中使用异常处理还是使用错误返回码处理，根据是否有利于**程序结构**来确定，并且异常 和错误码不应该混合使用，推荐使用异常，异常的处理效率比条件分支低，而且异常的跳转流程 难以预测

**34**.注意运算符的优先级，并用**括号明确**表达式的操作顺序，**避免**使用默认优先级

**35**.避免使用不易理解的数字，用有意义的标识来替代。涉及物理状态或者含有物理意义的常量，不 应直接使用数字，必须用**有意义的静态变量**来代替

**36**.数组声明的时候使用 **int[] index**，而不要使用 int index[]

**37**.异常捕获尽量不要直接 *catch (Exception ex)，应该把异常****细分处理***

**38**.集合中的数据如果不使用了应该**及时释放**，尤其是可重复使用的集合

**39**.在switch 中每个 case 语句都应该**包含 break 或者 return** A．不使用break或者 return从匹配的那个数据时一直执行下去； B．使用break执行完匹配的直接跳出switch程序块； C．使用return语句，整个程序执行到匹配的语句结束，不会执行switch之后的语句，除非有finally语句块

**40**.在运算中**不要减小**数据的精度

**41**.方法的参数名**不要**和类中的方法名相同

**42**.除了构建器外，不要使用**和类名相同**的方法名

**43**.不要显式导入 java.lang.\* 包，**自动导入**

**44**.**显式**初始化所有的局部变量

**45**.在进行比较的时候，把常量放在**右边**

**46**.使用集合时设置初始容量。也就是说使用集合不能初始为null，必须**new**一个对象

**47**.一个字符的连接使用**‘ ’**而不使用 “ ”，如：String a = b + 'McGrady'

**48**.不要使用不必要的布尔值比较，如：if (a.equals(b)), 而不是 if (a.equals(b)==true) **画蛇添足，多此一举**

**49**．尽量不要使用**内部类** 尽量不要使类可以**克隆** 尽量不要使接口可以**序列化** 尽量不要使用**友好**方法、属性和类

**50**.不要使用 Date.toString() ，Time.toString() 方法

**编码规范~云移**

@健壮性：功能准确前提下，是否考虑兼容空指针/数据异常/违法检查/边界值？  
@稳定性：模块之间是否松耦合？代码在各种环境各种操作下是否保证运行正确？  
@性能性：如果数据量很大是否能够支撑，并发性高是否考虑？多表连接的索引是否建立？是否可以改变方案使性能最大化？  
@安全性：接口调用是否有身份校验？敏感数据是否加密？  
@可扩展：如果需求改变了或者新增了是否好扩展，还是要全部推翻重来？  
@可复用：系统是否存在这样的工具类？会不会多个方法出现一样的代码？  
@可维护：代码是否按照编码规范编写？核心代码注释是否齐全？设计是否满足开闭原则？  
@可配置：是否能够动态配置，改变值可以不用修改代码？  
@高内聚低耦合：一个模块之内明确完成一个功能实现高内聚，模块之间要低耦合，模块下的类之间要多用组合少用继承。