**JAVA语言编码规范**

**编制单位：西部大区一部**

**编 制 人：王瀚贤**

**审 核：**

**批 准：**

**用友汽车信息科技（上海）股份有限公司**

**Yonyou Auto Information Technology (Shanghai) Co., Ltd.‘**

**文档更新记录**

| **编号** | **章节** | **变更类型** | **修订内容简述** | **修订日期** | **修订前版本** | **修订后版本** | **修订人** |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| 1 | 所有章节 | C |  | 2017-06-19 | / | 1.0 | 王瀚贤 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

\*变更类型：C——创建，A——增加，M——修改，D——删除

文档发送范围：

本部门员工。

**目录**

[1 代码书写风格规范 4](#_Toc443911557)

[2 命名规范 5](#_Toc443911558)

[3 注释书写风格规范 5](#_Toc443911559)

[3.1 注释总规则 5](#_Toc443911560)

[3.2 类的注释 5](#_Toc443911561)

[3.3 接口的注释 6](#_Toc443911562)

[3.4 函数的注释 6](#_Toc443911563)

[3.5 类属性的注释 7](#_Toc443911564)

[3.6 局部变量的注释 7](#_Toc443911565)

[4 优化性能方面的建议 7](#_Toc443911566)

# 代码书写风格规范

* 删除所有的无用代码以及注释。
* 程序块要采用缩进风格编写，缩进的空格数为4个。
* 必须给程序加注释。注释能够极大提高代码的质量。
* 代码的书写必须清晰、易读。

1. 适当的将代码分段可以使代码更加清楚。可以使用‘{’和‘}’将函数分成若干个小的单元。
2. 适当的加入空格或者空行可以使代码更加清晰。例如在函数的定义后面加入两个空行，在两个逻辑小代码单元间加入一个空行。
3. 所有的判断和循环语句都必须用括号扩起来，并且在较长（超过一屏）的判断或者循环语句的结尾应该有注释语句做出标识。
4. 建议遵循三十秒原则。如果另一个程序员无法在三十秒之内了解你的函数做了什么，如何做以及为什么要这样做，那就说明你的代码是难于维护的，必须得到提高。

* 在一个函数内代码的长度不允许超过100行，超过100行的函数，建议在不破坏原有算法的基础上拆分为多个函数。
* 程序中应用到的变量都需要在类或者方法的头部集中定义。
* 一行程序以小于80字符为宜，超长的语句应该在一个操作符之前折行，并在下一行加入适当的空格进行缩进。
* 保证一行代码只做一件事。
* 使用圆括号来界定操作的顺序。避免使用默认优先级。
* package 行要在 import 行之前，import 中标准的包名要在本地的包名之前，而且按照字母顺序排列。不同的包之间要加一个空格以提高代码的可读性。如果 import 行中包含了同一个包中的不同子目录，则应该用 \* 来处理。
* 使用统一的格式化代码。将‘{’放在所有者的后面，并且在下一行代码前加入四个空格的缩进。

if(XXX){

while(XXX){

xxx xxx

}

}

# 命名规范

* 各种标识符的命名要使用有实际意义的英文单词或英文单词的缩写，缩写要收录在该项目的简写词汇表中。切忌使用阿拉伯数字以及中文拼音进行命名。
* 在上一条的基础上，类名称要求大小写混写，首字母大写。
* 方法名称、局部变量名称要求大小写混写，首字母小写。
* 全局变量名称、常量名称要求全部大写。
* 参数名称与局部变量基本相同，区别在于参数名称的前面要加上冠词“a”或者“an”。
* 不允许出现名字隐藏，即局部变量、参数或者字段的名字，与另一个更大范围内定义的变量、参数或者字段的名字相同（或相似）。

# 使用工具规范

* 开发工具：Eclipse
* JDK版本：6.0（jdk1.6.0\_30）

# 注释书写风格规范

## 注释总规则

* 注释应该用中文清晰表达意思。应该能够使程序看起来更清晰，更容易理解。如果某一段程序不值得写文档，那么可能它是无效的代码。
* 注释要尽量简明，避免装饰性的、标语式的注释。
* 注释不但要说明做什么，还应当说明为什么要这样做。最好先写注释表明要做什么，再进行编码。
* 另起一行的注释，要写在被注释程序的上一行。

## 类的注释

* 类型的用途、目的。包括其它别人可能感兴趣的介绍。
* 已知的Bug。当然最好是修正所有的错误，但是有时可能暂时还没有办法修正错误，或暂时没有时间精力去修改。
* 开发和维护该类的历史列表。记录每一次修改的作者、日期、修改的内容。
* 列举类的各种稳定状态。说明调用成员函数使类的状态产生的变迁。
* 同步问题。
* 对主要的算法必须加以解释说明，主要的流程必须给出引导性的说明。

**标准格式：**

/\*\*

\* 描述类的功能、用途、现存BUG，以及其它别人可能感兴趣的介绍。

\* 作者：XXX

\* @version 最后修改日期

\* @see 需要参见的其它类

\* @since 从产品的那一个版本，此类被添加进来。（可选）

\* @deprecated该类从产品的那一个版本后，已经被其它类替换。（可选）

\*/

如果对已经版本话的类进行了修改，需要按照如下格式为每一次修改附加修改历史记录：

// 修改人 + 修改日期

// 修改说明

**范例：**

/\*\*

\* MessageDispatcher的主要功能是将消息源传来的各种消息，分发给相应的消息

\* 处理器进行处理。

\* 作者：张三

\* @version 2000/04/20

\* @see UFMessage, Event

\* @deprecated 从NC-1.0 版之后该类的功能被EventDealer取代，但为了保持向下兼容性，此类仍然被保留

\*/

// 李四 2000/05/21

// 添加了对帮助信息的处理函数sendToHelpManager(UFMessage msg)。

// 王小二 2000/06/02

// 修改了不能正确定位用户消息的错误。

## 接口的注释

* 接口的注释风格基本与类的注释风格相同。
* 在别人使用接口之前，必须了解接口所包含的概念。检验一个接口是否应该定义的简单方法是：你是否能够容易的描述接口的用途。
* 接口如何应当和不应当被使用。开发者需要知道该接口如何被使用，也希望知道该接口。
* 不能被怎样使用。

## 函数的注释

* 函数头注释必须包括：函数执行了什么功能，为什么要这样处理；函数处理过程中对对象的哪些属性可能进行更改；函数执行前后，对象的状态。
* 比较、循环等控制结构必须加注释。
* 在代码的功能并非一目了然的情况下，应当说明为什么要这样做。
* 局部变量必须加注释。
* 复杂难写的代码必须加注释。
* 如果一系列代码的前后执行顺序有要求，必须注释说明。

**函数头注释标准：**

/\*\*

\* 描述函数的功能、用途、对属性的更改，以及函数执行前后对象的状态。

\* @param 参数说明

\* @return 返回值

\* @exception 异常描述

\* @see 需要参见的其它内容

\* @since 从类的那一个版本，此方法被添加进来。（可选）

\* @deprecated该方法从类的那一个版本后，已经被其它方法替换。（可选）

\*/

**范例：**

/\*\*

\* 将消息转发给帮助管理器，显示对应的帮助内容。

\* @param msg ierp.sm.core.eo.UFMessage 消息的内容

\* @return ierp.lang.boolean 帮助管理器是否正确接收到消息

\* @exception java.lang.NullPointException 如果消息为NULL抛出此异常

\* @see ierp.sm.core.eo.UFMessage

\* @since 从NC-0.8版该方法被添加

\*/

## 类属性的注释

* 描述域的用途。使别人知道如何去使用它。
* 注释对变量的固定限制。例如 dayOfMonth 域就应该注明它的值在1-31之间。通过注释对域的限制，帮助你定义重要的事物规则，也使别人更加容易理解你的代码。
* 对于有着复杂事物规则的域，可以加入范例来说明。有时候一个简单的小例子，抵的上千言万语。
* 必须理清，并且清楚的注明并发问题。并发错误是很难在调试和维护阶段发现和排除的。

## 局部变量的注释

* 每行只声明一个局部变量，并且在行末加入注释。
* 注释的参考内容为说明该变量被什么、在那里、为什么使用。

# 优化性能方面的建议

* 若没有足够理由，不要把实例或类变量声明为公有，尽可能的私有化。
* 除非程序的框架需要，否则绝对不要因为性能的原因将类、方法定义为 final 的。
* 对于较长字符串的应用，要尽量使用StringBuffer对象，而不是直接使用String对象。
* 尽量使用 interfaces 来替代 abstract 类。
* exit() 方法只允许出现在 main 中，其他的地方不应该调用。
* 避免在 InternalFrame 组件中使用 AWT 组件。
* AWT 组件绝对不要用 JscrollPane 类来实现滚动。滚动 AWT 组件的时候一定要用
* AWT ScrollPane 组件来实现。

**附录**

**附录.1 匈牙利命名法前缀对应表**

**基本变量**

|  |  |  |
| --- | --- | --- |
| **变量类型** | **前缀** | **范例** |
| byte, short, int, long | n | nCount |
| Float | f | fTotal |
| Double | d | dSum |
| Char | c | cTemp |
| Boolean | b | bEndOfLine |
| array | ary | aryEmployees |

**常用工具类**

|  |  |  |
| --- | --- | --- |
| **变量类型** | **前缀** | **范例** |
| String | str | strName |
| Vector | vec | vecPackage |
| Hash | hash | hashContainer |
| StringBuffer | sb | sbSource |
| Object | obj | objData |

**常用控件**

|  |  |  |
| --- | --- | --- |
| **控件类型** | **前缀** | **范例** |
| Button | btn | btnOK |
| TextField | tf | tfName |
| TextArea | ta | taNote |
| CheckBox | ckb | ckbShowAll |
| RadioButton | rb | rbSex |
| List | list | listDepartment |
| ComboBox | cbb | cbbWorkCenter |
| Lable | lb | lbAddress |
| Table | tb | tbItems |
| Tree | tr | trAccount |
| ProgressBar | pb | pbMRP |
| Panel | pnl | pnlVoucher |
| ScrollPane | scp | scpPicture |
| TabbedPane | tbp | tbpMain |
| Dialog | dlg | dlgTip |
| Frame | frm | frmDesktop |
| Menu | mn | mnHelp |

说明：在可视化环境当中，由于VisualAge会在每一个控件的命名前加上小写的ivj前缀，所以在设置控件的BeanName属性时，需要将上表中的前缀的第一个字母大写。如：BtnCancel。

对于未定义标准前缀的类可以按照以下原则使用前缀：

* 1. 由一个单词构成类名，可以将这个类名全部小写（或者先缩写），作为前缀，在后面再加上表明用途含义的说明性修饰。如：

Employee employeeRetired

Mail mailUnread

* 1. 由多个单词构成的类名，可以取每个单词的首字母组成缩写组合（根据情况可以添加若干体现单词发音的字母），作为前缀。注意不要与标准缀重复。

WorkCenter wcAvailable

DefautMessageEvent dfmeReceived

**附录.2 常用动词表**

|  |  |  |
| --- | --- | --- |
| **汉语含义** | **英文单词或缩写** | **范例** |
| 做某事 | do | doQuery |
| 设置 | set | setName |
| 取得 | get | getName |
| 审核 | approval | approvalVoucher |
| 增加 | add | addEmployee |
| 修改 | modify | modifyContent |
| 删除 | delete,remove | deleteWorkCenter |
| 更新 | update,refresh |  |
| 查询 | query, search, find |  |
| 更新 | change | Chg |
| 返回、回复 | return | Rtn |
| 接收 | receiver | Recv |
| 跳转 | redirect |  |

没有全部列出，后续补充。

**附录.3 专业名词表**

|  |  |  |
| --- | --- | --- |
| **汉语含义** | **英文单词或缩写** | **范例** |
| 凭证 | voucher |  |
| 帐 | account |  |
| 科目 | Subject |  |
| 现金 | cash |  |
| 资产 | asset |  |
| 利息 | interest |  |
| 成本 | cost |  |
| 客户 | customer | Cust |
| 版本 | version | Ver |
| 操作员 | operater | OP |
| 信息 | information | Info |
| 细节 | detail | Dtl |
| 例外 | exception | Exp |
| 工厂 | factory | Fty |
| 条件 | condition | Cond |
| 属性 | attribute | Attrib |
| 描述 | description | Desc |
| 向量 | vector | Vct |
| 临时 | temp | Tmp |
| 业务 | business | Bus |
| 数量、数字 | number | Num |
| 订单 | order | Ord |
| 模块 | module | Mdl |