标准文档编号：HX-IT-BZ-00001

程序编码规范（JAVA 版）

华夏票联

2015 年 6 月

文 件 编 号	HX-IT-BZ-00001
文 件 名 称	编码规范（JAVA 版）
保密级别	内部使用
运 行 单 位	华夏票联

编 写 人	刘刚领	日 期	2015-06-08
审 核 人		日 期	
负 责 人		日 期	

版 本 号	修订日期	修 订 描 述
V1.0	2015-06-08	新建文件
V1.1	2015-06-09	修改包的命名规则，定义了开发层的命名规则

目 录

第1章　命名规范

命名总规则：

- 使用混合大小写的英文单词或英文缩写描述变量、类名、方法等。并且尽量使用该领域习惯的术语；英文缩写，必须统一定义，统一使用；
- 变量的命名一般遵循匈牙利命名法(首字母小写)；
- 避免超过 15 个字母的命名；
- 避免出现字母完全相同，仅大小写不同的命名；
- 不允许在命名的开头和结尾使用下划线等特殊字符。
- 为了保证命名的统一规范，建立一本常用命名词典，开发人员要严格按照命名词典对变量进行命名。

1.1 程序包（package）的命名规范

package 的名字全部由小写字母组成。

包名统一用代表模块贴切含义的英文单词或缩写，字母一律小写，每级包名的最大长度不能超过 8 位。包名为 4 级。

包名的第一级与第二级为固定名称: com.uticket

包名的第三级主要便于区分功能模块。如：公司管理为: company

包名的第四级代表系统架构分层。如：数据操作层为:dao、业务逻辑层：service、视图控制层为：action、实体层为：entity

整个包的目录示例如下：

com.uticket.company

--.dao

--.service

--.action

--.entity

所有的 DAO 层接口，都必须以 Dao 结尾，实现类以 DaoImpl 结尾

如:ICompanyDao,实现类为 CompanyDaoImpl

所有的 SERVICE 层接口，都必须以 Manager，实现类以 ManagerImpl

如:ICompanyManager,实现类为 CompanyManagerImpl

所有的 ACTION 层，都必须以 Action 结尾

1.2 Class 的命名

Class 的名字必须由大写字母开头而其他字母可以小写的单词组成。用第一个字母大写的英文正常语序准确描述类的含义,单词中间不允许出现下划线_。

如：MainFrame、MyCellRenderer

1.3 接口(interface)的命名

- 接口名的开头加上字母‘I’前缀。这可以明确区分接口和类，如：IRunable，ISingleton。
- 从第二个字母起，用首字母大写的英文单词描述接口。

1.4 类属性(property, member variable)的命名

- 变量的名字必须用一个小写字母开头。后面的单词用大写字母开头。如：

  private int vGap;

  private List fragments;
- 所有的域都是私有的，用并且仅用 getXXX 和 setXXX 等的存取函数去访问域,。包括成员函数中也应该这么做。这样起码做有两个好处，首先可以在函数中检查对域的赋值是否合法，其次更容易跟踪对域的访问。
- 必须初始化静态变量。并且尽量少用静态变量，除非对该变量可能发生的行为有充分的认识。
- 常量的命名全部使用大写。如：

  MAXVALUE
- Static Final 变量的名字应该都大写，并且指出完整含义。

1.5 方法的命名

- 函数名的第一个单词小写，后面的单词第一个字母大写；
- 第一个单词必须是动词，使函数的意义清晰明了；
- 存取对象的属性使用 setXXX()和 getXXX()函数形式；
- 访问布尔类型的属性使用 isXXX（）函数

范例：

```
openAccount()
save()
getFirstName()
isAtEnd()
setFirstName(String aName)
setAtEnd(boolean isAtEnd)
```

- 存取函数除了用来存取成员变量之外，还可以通过如下的手段来提高程序的柔韧性：

用存取函数访问常量。有些量在最初的事物模型中可能是常量，但当事物模型更改时，直接使用常量可能是灾难性的。例如：最初定义一周的工作日是 6 天，后来规则改变了。假如规则仅仅是将一周的工作日改为 5 天，并没有问题发生，但是当我们规定单周工作日为 6 天，而双周工作天数是 5 天，问题产生了。使用存取函数来访问常量将使我们的程序更加富有柔韧性。

1.6 方法的参数

使用有意义的参数命名，如果可能的话，使用和要赋值的字段一样的名字：

```
setCounter(int size){
        this.size = size;
    }
```

参数的名字必须和变量的命名规范一致。

1.7 局部变量(local variable)的命名

局部变量的命名和变量的命名规范一致。

- 仅在马上就使用它的地方，声明局部变量。这样可以避免别人滚动屏幕去看你的代码，并且清楚标明了变量的作用范围；如果你能遵循一个函数长度不超过一屏的原则，可以将变量全部都定义在函数的开始；
- 一个局部变量仅用来完成一项任务。一个局部变量被用于多个目的，往往会使你的代码难于理解，并且为错误埋下了祸根。反复使用一个局部变量可以节省一些内存，但是付出的代价确是高昂的。

1.8 数组的命名

数组应该总是用下面的方式来命名：

byte[] buffer;

而不是：

byte buffer[];

第2章　代码书写风格规范

- 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。

建议遵循三十秒原则。如果另一个开发人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。建议如果一个函数的代码长度超过一个屏幕，那么或许这个函数太长了。
- 一行代码尽量简短，并且保证一行代码只做一件事。那种看似技巧性的冗长代码只会增加代码维护的难度。保证不必左右拉动滚动条来阅读一整行代码，将会使你的代码更加可读，这一点包括注释；

- 使用圆括号来界定操作的顺序。不要让别人判断复杂的操作优先级。
- 使用统一的格式化代码。将‘{’放在所有者的后面。

```
if(XXX){
    xxx
        }
```

2.1 文档化

必须用 javadoc 来为类生成文档。不仅因为它是标准，这也是被各种 java 编译器都认可的方法。

2.2 缩进

缩进应该是每行 2 个空格. 不要在源文件中保存 Tab 字符. 在使用不同的源代码管理工具时 Tab 字符将因为用户设置的不同而扩展为不同的宽度。
如果你使用 UltrEdit 作为你的 Java 源代码编辑器的话，你可以通过如下操作来禁止保存 Tab 字符, 方法是通过 UltrEdit 中先设定 Tab 使用的长度是 2 个空格，然后用 Format|Tabs to Spaces 菜单将 Tab 转换为空格。

2.3 {} 对

{} 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.
} 语句应该缩进到与其相对应的 { 那一行相对齐的位置。

2.4 括号

左括号和后一个字符之间不应该出现空格，同样，右括号和前一个字符之间也不应该出现空格．下面的例子说明括号和空格的错误及正确使用:

CallProc(AParameter); // 错误

CallProc(AParameter); // 正确

2.5 Package/Imports

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。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。

```
package com.uticket.company.dao.da;
import java.io.*;
import java.util.Observable;
import com. uticket.tools.pub;
```

这里 java.io.* 使用来代替 InputStream and OutputStream 的。

2.6 main 方法

如果 main(String[]) 方法已经定义了，那么它应该写在类的底部。

第3章 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。

3.1 版权信息

版权信息必须在 java 文件的开头，例子如下：

/**

* Copyright 2004 Beijing xxx Tecnology Co. Ltd.

* All right reserved.

*/

其他不需要出现在 javadoc 的信息也可以包含在这里。

3.2 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
*/
```

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：

```
/**
 * DaSupplyPower.的主要功能是将显示负荷计划曲线。
* 作者：张三
* @version      2003/04/20
* @see          PubCalendar MsgBox
*/
// 李四  2003/05/21
// 添加了更改图形显示方式的处理函数 changeShowFloag()。
// 王小二  2003/06/02
// 修改了日期不正确的错误。
```

3.3 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。

3.4 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那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(每个参数的说明占单独一行)
 * @return      返回值
 * @exception  异常描述
 * @see       需要参见的其它内容
*/
```

范例：

```
/**
 * 根据传入的项目名称 ID，得到项目名称的中文名称。
 * @param       String powerid  项目名称 ID
 * @return      String  项目名称的中文名称。
 * @exception   java.lang.NullPointException  如果得不到项目名称的中文名称抛出此异常。
 * @see       无
*/
```

3.5 类属性的注释

- public 的成员变量必须生成文档（JavaDoc）
- 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；
- 注释说明该变量的含义；

第4章 附录

4.1 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

4.2 常用动词表

汉语含义	英文单词或缩写	范例
增加	add	
删除	del	
修改	modify	
保存	save	
取消	cancel	
最前	first	
上条	prior	
下条	next	
最后	last	
刷新	refresh	
查询	query	
统计	sum	
浏览	browse	
返回	return	
退出	exit	
放弃	abort	
打印	print	