**项目编码规范**

# 编码规范

## 1.格式与命名规范(Formating and Naming Conventions)

1. 一般直接使用Eclipse的自动格式化功能。
2. 换行:每行120字符以上--因为现在屏幕已大为宽广。
3. 括号:if,for,while语句全部使用括号包围。
4. 命名规则:  
   不允许使用汉语拼音命名  
   避免使用下划线(静态变量除外)  
    接口尽量采用"able", "ible", or "er"，如Runnable命名，尽量不采用首字母为I或加上IF后缀的命名方式，如IBookDao,BookDaoIF。(II)

## 2.注释规范(Document Convertions)

### 2.1 注释类型

1. JAVA DOC注释  
   /\*\* .... \*/
2. 失效代码注释  
   由/\*... \*/界定，标准的C-Style的注释。专用于注释已失效的代码。
3. 代码细节注释  
   由//界定，专用于注释代码细节。  
   注意：即使有多行注释也仍然使用//，以便与用/\*\*/注释的失效代码分开。

### 2.2 注释的格式

1. 注释中的第一个句子要以（英文）句号、问号或者感叹号结束。Javadoc生成工具会将注释中的第一个句子放在方法汇总表和索引中。
2. 为了在JavaDoc和IDE中能快速链接跳转到相关联的类与方法，尽量多的使用@see xxx.MyClass，@see xx.MyClass#find(String)。
3. Class必须以@author声明作者，体现代码责任。但不需要声明@version与@date，由版本管理系统保留此信息。(II)
4. 标识(java keyword, class/method/field/argument名，Constants)在注释中第一次出现时以 {@linkxxx.Myclass}注解以便JavaDoc与IDE中可以链接。(II)

### 2.3 注释的内容

1. 可精简的注释内容  
   注释中的每一个单词都要有其不可缺少的意义，注释里不需要自动生成一堆空的标签，如空的@param name，空的@return。
2. 推荐的注释内容  
   对于调用复杂的API尽量提供代码示例。(II)  
   对于已知的Bug需要声明，//TODO 或 //FIXME 声明:未做/有Bug的代码。(II)  
   Null规约:  如果方法允许Null作为参数，或者允许返回值为Null，必须在JavaDoc中说明。否则方法的调用者不允许使用Null作为参数，并认为返回值是Null Safe(不会返回NULL)。

## 3.编程规范(Programming Conventions)

### 3.1 基本规范

1. 当API会面对不可知的调用者时，方法需要对输入参数进行校验，如不符合则抛出IllegalArgumentException，建议使用Spring的Assert系列函数。
2. 因为System.out.println()，e.printStackTrace()仅把信息显示在控制台，因此不允许使用，必须使用logger打印并记录信息。
3. 在数组中的元素(如String [1])，如果不再使用需要设为NULL，否则会内存泄漏。因此直接用Collections类而不要使用数组。
4. 在不需要封闭修改的时候，尽量使用protected 而不是 private，方便子类重载。
5. 变量，参数和返回值定义尽量基于接口而不是具体实现类，如Map map = new HashMap();
6. 用double 而不是Float，因为float会容易出现小数点后N位的误差。

### 3.2 异常处理

1. 重新抛出的异常必须保留原来的异常，即throw new NewException("message", e); 而不能写成throw new NewException("message")。
2. 在所有异常被捕获且没有重新抛出的地方必须写日志。 --TODO
3. 如果属于正常异常的空异常处理块必须注释说明原因，否则不允许空的catch块。

### 3.3 JDK5.0规范

1. 重载方法必须使用@Override，可避免父类方法改变时导致重载函数失效。
2. 不需要关心的warning信息用@SuppressWarnings("unused"), @SuppressWarnings("unchecked"), @SuppressWarnings("serial") 注释。