1. 目的

为了使软件开发过程顺畅，保证软件质量，于是有了这份开发用的“轻”规范文档。

2. 概述

* 项目以功能模块来划分分工
* 数据库不能随意修改，本地数据库须统一
* 有标准的Java代码风格，有良好习惯
* 时刻考虑代码的可复用性
* 每个人需要有每周项目进度报告

3. 代码规范

3.1 有关命名

尽量保证能够通过变量名得知变量的含义

3.1.1 包命名

* 包命名采用全**小写**命名
* 通过唯一域名+组件(功能模块)来命名

3.1.2 类命名

* 类命名采用**Pascal命名法**
* 大写字母开头，各个单词首字母大写

3.1.3 方法命名

* 方法命名采用**Camel命名法**
* 小写字母开头，各个单词首字母大写
* 属性的getter和setter方法尽量使用自动生成，并放在程序的后面。
* Boolean类型的属性的get方法应形如isProperty() (但不排除使用如性别男女：getIsMale或 getIsFemale)

3.1.4 变量命名

* 采用**Camel命名法**
* 小写字母开头，各个单词首字母大写
* 特有大写缩写词汇保持大写如：SQL
* 变量名字不宜过长，可适当采用缩减英文元音字母来缩短长度
* 假如缩短后名字重复，可以保留其中一个的部分元音字母

3.1.5 常量命名

* 采用**全大写命名法**   
  所有字母均大写

3.1.6 页面文件命名

* 采用**全小写命名法**
* 所有的字母均小写，单词之间以下划线’\_’分隔
* 展示页面，按照名词+描述，如：news\_list.jsp ; 或名词（文件夹）+ 描述,如：news/list.jsp
* 操作页面，按照名词+动词命名，如：news\_add.jsp; 或名词（文件夹）+ 描述,如：news/list.jsp
* 按照模块从大到小命名，如：news\_order\_add.jsp ; 或名词（文件夹）+ 描述,如：cms/news/order\_add.jsp

3.1.7 资源文件命名

* 采用**全小写命名法**
* 按照前缀+模块+描述+状态命名，如：btn\_main\_login\_pressed.png
* (不一定完全包含如上4个部分，但是要依照顺序命名)

3.2 有关注释

团队成员都应该形成良好的写注释的习惯，方便以后阅读，以及为了后期生成可读性良好的Java Doc

3.2.1 程序文件头注释

应该包含如下：   
\* 文件描述   
\* 作者   
\* 版本   
\* 创建日期时间   
\* 修改日期时间   
\* 参考信息

**提前设置好文件的模板Template**   
如以下模板：

/\*\*

\* Description:

\* Author: KevinLee

\* Version: 1.0

\* Create Date Time: ${DATE} ${TIME}.

\* Update Date Time:

\* @see

\*/

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8

3.2.2 方法头注释

一般在写完一个方法后使用快捷键生成一个块注释，IDE会自动帮我们写入一些信息。   
应该包含如下信息：   
\* 方法描述 Description:   
\* 参数信息 @param   
\* 返回信息 @return   
\* 异常信息 @Exception   
\* 参考信息（可选）@see also //指定一个类或者方法（通过类后面加#选择方法）   
\* 笔记信息（可选）Note:   
如以下模板：

/\*\*

\* Description: 返回一个“Hello”字符串

\* @param str 一个字符串

\* @return 返回一个字符串

\* @throws Exception 抛出一个异常

\* @see com.lidengju.entity.Person

\* Note: Nothing much.

\*/

public String sayHello(String str) throws Exception{

str="Hello";

return str;

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12

**注意：方法里面不要使用块注释**

3.2.3 关键点注释

应该包含如下信息：   
\* 一些程序关键的地方   
\* 一些程序不易读的地方   
\* 编写代码过程中遇到问题的地方   
\* 需要提示读者的地方

注释应该写得少而易懂   
若修改了文件，可以加上修改人的信息，和修改日期（但不强制添加修改人和日期 )。

4. 格式规范

4.1 缩进

应注意使用format来格式化代码，使用Tab键来缩进代码，相当于4个空格。

禁止使用全选+format来格式化整个文件（会影响字符的连接格式、开发人员特定的语句行缩进等）

4.2 换行

* {}花括号应该另起一行，左花括号与方法名、类名在同一行。(除了数组初始化时的花括号)
* if、while等语句，假如体内只有一句代码也不要省略{}，为了方便以后的增删
* 字符串过长考虑拆分成多行，并添加适当的前后空格(如：SQL断行,XML断行);

4.3 对齐

* {}括号等应该对齐
* 类和方法的块注释必须紧贴类和方法
* 单独起行的//注释必须对齐被注释语句

5. 写在后面

希望各位成员遵守这份开发规范文档，养成良好的开发习惯