**社区用户有偿互助平台编码标准**

作者： 董振杰 张伟腾 苏仲 邓岳英

版本：         0.1

日期：      2018-11-29

**修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期** | **修订版本** | **修改描述** | **作者** | **审核** |
| 2018-11-29 | 0.1 | 草拟 | 小组全员 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目录**

1. 概述... 5

2. 代码规范... 5

2.1 有关命名... 5

2.1.1 包命名... 5

2.1.2 类命名... 5

2.1.3 方法命名... 6

2.1.4 变量命名... 6

2.1.5 常量命名... 7

2.1.6 资源文件命名... 7

2.1.7 其他命名... 7

2.2 有关注释... 8

2.2.1 程序文件头注释... 8

2.2.2 方法头注释... 8

2.2.3 关键点注释... 9

3. 格式规范... 9

3.1 缩进... 9

3.2 换行... 10

3.3 对齐... 10

**1. 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；
* 根据老师安排，每个人在需要时应提供每周项目进度报告；

**2. 代码规范**

**2.1 有关命名**

       总体原则上，变量名与方法命名时应该遵循良好的命名习惯，尽量使得命名能够体现变量或者方法本身的含义。

**2.1.1 包命名**

* 包命名采用全小写命名；
* 通过功能来命名；

**2.1.2 类命名**

* 通过功能来命名；
* 类命名采用Pascal命名法；
* 小写字母开头，各个单词首字母大写；

**2.1.3 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()；
* 使用getXXX()形式和setXXX()形式进行获取数据和设置数据的基本方法命名；

**2.1.4 变量命名**

* 特有大写缩写词汇保持大写如：SQL
* 尽量采用全命名方式，名字确实过长时，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；
* 类内部变量命名时，需要带上m开头，表示此为类内部变量成员；

**3.1.5 常量命名**

* 参照变量

**3.1.6 资源文件命名**

* 控件在命名时保持与layout中id一致命名原则；
* 采用全大写命名法，有意义的单词之间使用"\_"进行分割；
* 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；
* 按照分类+功能+所属页面等命名，如：layout\_main.xml，item\_bluetooth\_device.xml,view\_switch\_button.xml,dialog\_add.xml等；
* 图标文件默认使用ic\_开头，例如ic\_logo；
* 控件样式文件等以控件缩写开头，描述控件目标，例如btn\_ background \_light.9.png；
* 原则上采用全小写命名法，并采用下划线分割法；
* 控件id在冲突的情况下需要加入页面限定，例如：tv\_username\_activity\_main等；

**3.1.7 其他命名**

各个控件以首字母缩写为基本原则，个别控件可根据实际情况使用全命名方式，但必须小写开头，命名前缀一览；

其它情况下需要提出问题进行商讨解决；

**2.2 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。

**2.2.1 程序文件头注释**

应该包含如下：   
\* 文件描述 （Description）：描述此类的作用；  
\* 作者 （Author）：创建者或者修改者名；  
\* 版本 （Version）：创建或者修复时的编号，需要自行在bug管理系统中创建bug  号，使用bug号进行命名（若无bug管理工具的临时办法：如无bug号，从1开始，修改时依次增加）  
\* 日期（Date）：创建或者修改时的日期，使用“-”进行年月日分割；

   \* 记录（Record）：创建或者修改的工作内容描述；

       提前设置好文件的模板Template， 模板以下：

/\*\*

 \* Description:

 \* Author          Version        Date           Record

 \* KevinLee        1              2017-11-7      版本创建

 \*/

**2.2.2 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   
       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

 \*/

public  String  sayHello(String  str) throws  Exception{

    str="Hello";

  return  str;

}

注意：方法里面不要使用块注释

**2.2.3 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；

       \* 注释易懂 ；

**3. 格式规范**

**3.1 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。折行使用120字符宽度；

**3.2 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；
* {}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；

**3.3 对齐**

* 同一层次的语句应该左对齐；
* If，while等语句具有的子语句应该缩进；