

# 步客小队代码规范

## 1. 注释

- a. 逻辑关键处，不易理解处，使用“//”注释，不只是简单的翻译代码，需要讲清逻辑
- b. 类、接口、方法使用 “/\*\* \*/” java doc 标准注释。
- c. 每个类和接口都要给概述，每个方法都要有功能描述

## 2. 命名

命名时注意，要能准确传达含义。如有命名中英文单词不熟悉，或者没想到更好的命名方式，在后面用“//”注释方式注释一下。

## 3. 布局格式

- a. 采用下图的大括号缩进方式

```
if (.....) {  
    return true;  
} else {  
    return false;  
}
```

- b. 同一方法中，不同逻辑块用空行分隔。

- c. 每一行代码，为了提高可读性，适当使用空行，如

```
int a = 2;  
  
public int add(int a, int b) {  
    return a + b;  
}
```

## 4. 类定义的逻辑组织

- a. 成员变量声明
- b. 构造方法
- c. public 方

d . protected 方法

e . private 方法

5. 长句断行

```
return this==obj
    || (this.obj instanceof MyClass
        && this.field == obj.field);
```

6. 出现复杂决策,利用新的 boolean 变量或者新的布尔函数或者决策表来简化

```
boolean allDataReaded= (atEndofStream) &&(error!= inp
boolean validLineCount =( MIN_LINES<=lineCount)&& lineCc

if ( allDataReaded && validLineCount && ( ! errorProcessing)
```

7. 减少代码中的硬编码,采用变量值来替换,提高可修改性。必要时可以使用

```
static final int xxx
```

8. 防御式编程,注意每个获得变量有无初始化或者是否为 null。

可以自定义 exception 类,抛出异常有利于查找 bug。

也可以利用断言 assert 和 assertEquals。

**特别要注意这一点,因为调用方法极有可能是别人写的,返回值正确与否你并不知道,所以一定要先判断。**

9. 当一个方法的代码量出现一个屏幕不能全部显示的时候,考虑把内部一些逻辑的实现编写成新的函数。

10.类的方法及成员变量权限注意 public 和 private 的区分。成员变量尽量使用 private, 修改利用 get 和 set 方法。

11.每个人写的代码一定要自己进行过单元测试才能提交到 git 上。

一些话：

代码规范有啥意见大家自己提出来，一起修改。

我看了一下往年课件，集成测试的东西主要在下一节软工课讲。上面关于 driver，stub，mock 在测试中用法，结构讲的感觉还是挺清楚的。大家可以先看看。我的想法是最近几天把它搞定。

大家上课一定要认真听一下，听的同时把讲的东西和咱们大作业结合起来，想想哪些需要改，哪些需要注意，**有选择的记录一些觉得重要的**。每个人听课的注意到的和理解肯定会有所差别，最后讨论一下，这样也减少了理解上的错误。

**在每次开会之前考虑一下今天开会的任务需要怎么完成**，如果没什么头绪可以自己看 ppt 或者书或者问问别的同学。带着点想法去开会，这样开会效率才高，讨论出的东西才多。

任何人都不要有“有别人考虑 我只要听就行了”的想法。（没啥意思，有则改无则勉）

毕竟，大作业是一个小组的，缺不了任何一个人的付出。