

代码编写原则

一、写在前面

本原则为在线报修 app 项目组整理编写。

为了不让因代码杂乱冗余而心生厌恶,为了不让过因代码晦涩难懂而抓狂,为了不让因代码低级的逻辑错误而愤慨,为了不让因代码结构不合常规而需要到处查找,为了不让因看到几百甚至上千行代码的方法而望洋兴叹,为了不让因代码缺少注释而猜测以及花很多时间去理清前后逻辑,在项目开始之前先制定一个浅显易懂又十分重要的代码原则,以供项目组成员参考和阅读,来提高开发的效率和保持团队的和谐与愉快。

请记住代码不是一次性的,需要重复的修改和重构,我们是在为未来写代码!

二、原则内容

● 写干净整洁的代码

去除没有用到的类引|看到一大片的因为类引用没有用到而报的警告信息简直就是灾难,空间和视觉都没有得到尊重。

记得格式化代码!看到一大片杂论无章,连基本的对齐都没有的代码也是很大的灾难,视觉脑力都会受到挑战。

不要吝惜废弃的老代码！有些人对待已经废弃的老代码比对待自己的女票还宝贝，大量运用注释来保留，随着代码的演变占用非常大量的空间。如果那段代码非常精妙，舍不得删，那么请把它移到你的私人代码库。

请不要写冗余无用的代码。if(true)之类的代码块完全不应该出现，用空行隔开该段代码是更好的选择。

请合理运用空行。空行可以用来隔开相对独立的代码块，有利于阅读和理解。但是不要使用超过一行的空行，对空间，别太奢侈了。

请不要在两个地方出现完全相同的代码，你总是可以想办法重用的，不是吗？

命名类，方法，变量慎用简写，除非大家都公认。全称都不一定看得懂，简称就太高估后来人的智商了！请问 qrbz 代表什么，可能火星人知道吧！

把所有的类变量放到最前面，如果比较多请按用途分组排列，不要把变量散落在大江南北，别人找的真的很辛苦！

拆分大的类，大的方法，如果你的类有一万行，如果你的方法有一千行，别人真的会恐惧，是恐惧！

● 高效运用注释

非通用方法的 public 方法都需要注释，你总不会要别人读你的代码才能调用你的代码吧！即便你的英文非常好，命名也非常规范，你总不能期待别人都英文好，理解都一样吧。

为不容易理解的类变量注释。类变量特别是私有的类变量没有人要求注释,但是为了能够快速的了解你表示的是什麼,还是写点什麼吧!

注释代码段,注释逻辑选择。上面提到运用空行分割开逻辑相对独立的代码,那么请在空行的下一行也写点下面代码段要干什么的语句吧。如果有 if else 等逻辑选择的时候,麻烦也花几秒钟写上判断的依据和结果,逻辑难懂且关键,你懂的!

- **不断学习, 不断思考, 不断实践, 更上一层楼**

遇到不懂的, 请先 google, 一定要学会, 不然就一直不会了!

看别人代码时要汲取好的方法和技巧。

接触一项技术要深入了解和实践, 请问你做过的系统你现在都可以从零开始搭建起来了
吗, 我的意思是架构搭建哦!