

Question3:如何评价代码质量？好的标准是什么？

如何评价代码质量？

WTF/min是衡量代码质量的唯一标准，不管代码的格式、注释、写法如何，如果让人看了抓狂就不是好代码。所以我认为易读性是衡量代码质量的首要标准。另外好的代码应该还要具有以下特性：

- 正确性：满足特定的任务需求；
 - 可靠性：软件系统在一定的时间内无故障运行的能力；
 - 性能：能满足极端情况下也能稳定提供服务，比如双十一的阿里服务器的超高并发处理；
 - 安全性：漏洞要少；
 - 可重用：可以复用代码；
 - 可维护；可移植；可测试；方便升级迭代。
-

代码好的标准是什么？

整洁的代码是好的代码的必备属性

整洁的代码是：

- 优雅且高效，直截了当；
- 减少依赖，一次只做好一件事；
- 是简单直接的；
- 是易读，可维护，可方便进行单元测试的；

整洁的代码应该具备一下五个条件：

1 有内涵的命名

好的命名应该具备：

- 名副其实：做到不用解释，读者自然就知道是啥意思。代码写来不是只给机器看的，更重要的是人要阅读，要对人友好。
- 避免误导：误导性的命名很致命，如果直接把人带偏了，这样理解起来根据挑战。

2 意图明确的注释

- 表明意图：想做什么，为什么这么写，实现的主体功能；
- 警示：对代码关键部分作必要说明，以及他人复用此代码需要注意的地方；
- 阐释输入输出含义，以及关键参数的含义；

3 短小精悍的函数

- 单一职责：一个函数应该只做一件事，而且这件事应该能通过函数名就能清晰的展示；
- 函数参数：函数的参数越多，组合出的输入情况就愈多，需要的测试用例也就越多，也就越容易出问题。

4 统一的团队代码格式

- 垂直格式：源文件最顶部应该给出高层次概念和算法，细节应该往下渐次展开；
 - 垂直距离：
 - 变量声明尽可能靠近其使用位置。
 - 概念相关的代码应该放到一起，相关性越强，彼此之间的距离就该越短。

- 若某个函数调用另一个，就应当把他们放在相近的地方。
- 垂直顺序:
 - 最上面的代码应该是最抽象的，底部细节应该在下面实现。这样就可一先大致建立起来一个框架，最重要的概念在最前面，慢慢展示出来细节的东西。
- 横向格式：一行代码要短，利于理解，所以应该尽力保持代码行短小(在30个字符以内)
- 缩进：统一的缩进格式，乱缩进的代码看起来是真的难受。
- 团队规则：代码风格统一，一组开发者应当认同一种格式风格。

5 明确清晰的类

- 类的组织：
 - 类应该从一组变量列表开始。先列出来公共静态常量，然后是私有静态变量，以及私有实体变量；
 - 公共函数应跟在变量列表之后，最后再是私有函数。
- 类应该短小：
 - 类名应该精确。类的名称应该描述其全责，一个类应该只有一个全责；
 - 类应该只有少量实体变量。类中的每个方法都应该操作一个或多个这样的变量。