

不要为了架构 而架构，每一件事都有利有弊，首先你需要弄明白 你需要什么，

1：架构是在 有一定规模的情况下 才需要的，当就几行代码 ，例如：判断某个变量不为 1，则置 1，还是 直接 置 1，

2：c 语言 为了提升性能，是没有做边界检查的，

3：c++ 类里面有个 属性，外部是需要多次使用的，面向对象思想 这个属性是需要设置成 private

附加 两个函数 set 和 get，我以前总认为 不如直接将 这个属性设置成 public，设置成 private 的用途：1：隐藏；2： set 和 get 函数中 可以实现一些处理，例如判断 set 是否合理，不会导致 set 了一个 乱值，但是这样 就牺牲了性能，

4：有些代码写的很搞笑，一个变量 被多次 传递，每次传递都判断下 该变量是否为空，考虑：1：该变量是否会改变，会的话 需要判断，不会不需要判断