### 我好像有点明白马丁·福勒说的一句话

你问我马丁·福勒是谁？嗯，我只能告诉你我也只是听过他的大名以及他写过的一本书：《重构》。刚才刷微博看到有个叫Codeif的网站被访问爆了，好吧这只是个帮助程序猿命名变量的网站！网站首页挂着马前辈说的一句话：计算机科学的两个唯一难题是缓存失效和命名。嗯，我承认看到这句话我就已经压抑不住出来装x的冲动了，所以就有了这篇文章。

马先生说的这话是啥意思，百度出的是一堆blabla不知所云的因为所以，看样子装x不成啊。于是只能自个儿脑补，以为已参透其中几分深意，，哈哈～说说后半句：命名。所谓命名么就是给一类事物归类，说的高大上一点就是抽象，知道俺们程序猿为什么老纠结于给一个变量命名了吧，不是英语不好，不是英语不好，不是英语不好！是抽象能力差啦。我们码代码的时候到处都存在命名，定义变量，定义方法，定义模块，这些其实都是一种抽象行为。我们往往得不到一个好的抽象结果，代码一膨胀，问题就显现出来了。这可能是因为没有理清各个部分的职责，这个方法该干嘛，那个模块做什么用，各自有没有清晰的职职责范围。还有可能和写代码的人的随意性有关，方法，接口胡乱定义，没有明确的注释说明，导致使用接口的人无所适从。之前看过的一些书里或多或少讲到前辈们的经验总结，这些经验看似微不足道，但却很实用，列举如下：

1.类默认应该只在模块内可访问；

2.方法默认应该是私有的；

3.方法要名副其实，并且只做一件事；

4.方法参数应该泛化（就是参数类型尽量用高层的父类）；

5.方法返回值应该具体化（就是返回类型尽量用低层子类）；

6.方法的参数应该是精准没有歧义的。就是需且仅需要这些参数；

关于对外公开的业务服务：

1.服务应该是饱满的。服务需要做好入参及业务规则校验，如果执行成功，那一定是符合业务规则的；

2.服务是细粒度的。服务只做一件事，一个服务应该对应一个不可分的事务；

3.服务需要详细的文档说明，如果不能做到使用者一目了然或者有歧义，就应该修改服务的定义；

说到这里，是不是觉得有点跑题了，，跟“命名”有毛线关系，好像是这样，但是回头想一想我们需要做的所有事情就是在给一个事物下一个准确无误，明确没有歧义的定义，不就是命名么，，，（怎么感觉听着有点扯淡），不过anyway啦，不要在意这点细节。其实重点就在于抽象。人类在计算机科学上最伟大的创造在于：将人的思维转化成了高低电压，这是抽象的力量。不知道又是哪位前辈讲过的一句话：计算机上的一切问题都可以通过增加一个中间层来解决，这个中间层又是抽象。