

作者：MyBatis 中文官网

网址：<http://www.mybatis.cn/>

更新时间：2019 年 3 月

发布介绍：

本系列内容会不定时更新，最新内容，请关注 MyBatis 中文官网。

本系列内容，版权属于 MyBatis 中文官网，任何人不得复制和转载。

前言

弹指十年，终成此文，有名其曰：沉思录。

本系列仅是一家之言，只供参考，望读者能明鉴，取其用，弃之误。

因个人原因，武断误信，误入歧途，以致劳心伤神，事倍功半，概不负责。

最后告知：请不要过来找我背锅。

第一节：如何提高四六级的英语听力

首先要明白一个重要道理：听力的考察，实际上考察的还是说的能力。如果考察口语水平，其实教育部和考试院的投入成本很大，而且不容易设立公平和公正的考核标准。

明白了上述道理，下一步就好说了，务必要多读多背。比起多读我更乐意去多背，我感觉熟练地背诵一篇文章的效果，顶上读一百篇文章的效果。

有学生给我说，他考了一次托福，一个感受就是自己的听力能力很差，一个是辨音能力比较弱，一个是长时间听时容易疲劳分心。

听力的提高在于背书，听力考察的不是听，而是说的能力。之前给他说过的道理，好像他并没有听进去。虽然他在听听力的时候都会去复述相关内容，但是我仍然建议他要背一定数量的文章。我之前背了 200 多篇文章，感觉对听力很有帮助。

英语听力理解要求的时间非常短，必须在极短的时间反映出对说的是什么意思，要想形成这么快的反应能力，必须走背诵这条路了。之前我也曾使用过复述和多读的方式，但是感觉效果不大，直到我改变了学习方法，以背诵为主之后才出现了明显的提升。多背是根本，多读是辅助，不要搞的本末倒置了。任何想提高英语四六级的同学都可以试一试多背这个学习方

法。

第二节：如何学习数学？

群成员提问：群主有木有学习数学的经验？

群主回答：

目前来看，我这辈子能走到今天靠的就是数学。你这个问题问到我身上，那是真问对人了。在数学方面，怎么学，我算是研究的很透彻。但是我要讲的话，估计得讲几天几夜都难以讲完。

有的人能学好数学，但是他说不出来，怎么能学好；有的人学不好数学，他说的其实都是错误的。我属于那种能学好，也知道怎么学的。

把考试大纲拿过来，把需要了解和理解的知识点全部扔掉，把需要掌握的知识点，往死里学习，往死里练习，把这些知识点倒背如流都不为过。拿到考卷，自然就知道怎么做了，就是碰到不会做的题，想一想那些知识点还没有用到，套进去就可以了。

考卷有个要求：必须覆盖 90%以上的知识点（需要掌握层次），这个规定其实就是一个最大的漏洞。

这个是考试数学的方法。另外，要想在数学上有创新，那还得靠悟性，可不是刷题就刷出来的。

数学，一般都是三十岁之后才能学好的科目。

任何一种经验都是一种绝活，掌握到火候才能发挥出巨大的威力，没有掌握到火候，顶多算是看看热闹而已。

第三节：读源码？NO，别被人忽悠了。

培训机构怎么吸引眼球呢？分析源码啊。公众号怎么涨粉呢？分析源码啊。当真读源码的方式就是最好的学习途径吗？非也。

我不认可过分强调读源码的学习方式。我觉得：上来就读源码，这种学习方式是大错特错的。一定先熟悉使用文档，这才是关键。大多数人读完源码，不能说没有提升，但是相比投入时间，产出比太低了。培训机构，都靠挖掘源码当噱头、当卖点，我们不要上他们的当。

出去面试，一个人数据结构和算法厉害，一个人读源码多，肯定面试官选择第一人。学习技

术，我们要明白那些是重点，那些是非重点。

学习技术，一定要把使用文档搞清楚，文档本身也是源码，是源码的更高层封装，我们读这些使用文档，其实也是在读源码的。而且，别人写好的文章，我们读一遍，相当于借力，这样对我们的成长速度很有帮助。

如果框架学习的很好了，而且有业余时间，也有兴趣，那么可以看看源码。

我读 mybatis 源码花费了很多时间，忽略了使用文档，感觉效果不好。后来我学习 zk 的时候，不去读 zk 的源码，但是我搜遍了 zk 的介绍和使用，我发现 zk 的使用远比 mybatis 更有感觉。

虽然我也有时候读源码，但是我不鼓励大家去读源码，而且不建议上来就读源码，另外一定还要控制住量，否则的话，变成了码海战术了，肯定是个大弯路。

别说初学者了，中高级开发者也很容易犯过分强调读源码的毛病。因为我们的环境无时无刻有人在误导你，例如培训机构，公众号等等。做开发真不容易啊，到处都是套路！一不小心，你就会被误导，被焦虑，被灌鸡汤了。

最后，强调一点，不是反对读源码，而是要明白做事要讲究先后顺序。

补充一点吧，比读源码提升更快的学习方式应该是写工具，写一个属于自己的东西。这东西很实在，拿得出手，贴简历上是亮点，而且能锻炼人。比起读源码更难，有难度自然提升才更快。

第四节：外包要不要去？

某群成员的观点如下：

我总结下，像我这种普通本科的：外包的好处就是能接触到一些优秀的人，甲方有些人可能能力不咋地，但是他们绝对是认真负责的。这一点很值得学习。坏处：工作毫无体验可言，没有归属感。

站长的观点是：任何岗位不要待的超过 2 年，最好 1 到 2 年左右即可，可以去跳槽或者内部岗位轮换。除非非常好的岗位，不再此列。同一岗位待的过长，就变成重复工作了。重复工作是技术人的最大杀手。

立足于在这个大的原则下，去不去外包不是一件纠结的事情，反正到哪里都会走人的，顺心就多待，不顺心就少待。

跳出舒适圈，说白了，就是要多挑战自己，多跳槽也是挑战自己。

重复工作一两年，没有成长和突破，当时可能感觉无所谓，等年龄大了就会明白了，一两年真的很关键的，在生命中其实很重要的。一两年有的人可能提升很快，一两年有的人几乎就

没有什么发展，等后面的新人涌入，那还有那么的机会等着给你啊，那个时候就憋屈了。

第五节：编程的艺术

我一直觉得：只做 Java 的人，永远成不了 Java 高手。下面内容是顿悟出来的，是基于 Python 语言的特性：函数，也是一种对象。立足这个思想，将 Java 里面函数，升华成内部类，而且让原功能带上线程池+异步执行的特性。具体如何实现呢？请看下文：

类里面包括两种物质：属性和函数。在某些编程语言里面，将函数也当做一个对象。从这个角度来看，函数就像一个内部类。

把函数当做内部类，这种想法别出心裁，挺新颖的，但是并没有特别使用之处。可以将其彻底升华一下，从而将这种想法发挥出巨大威力，以提升代码的设计能力和思想境界：

(1) 函数变成内部类，而这个内部类 implements Runnable，给函数的执行增加了异步的功能。这个内部类，可以是普通的类，也可以看做一个事件类

(2) 函数变成内部类之后，之前函数的参数变成内部类的属性

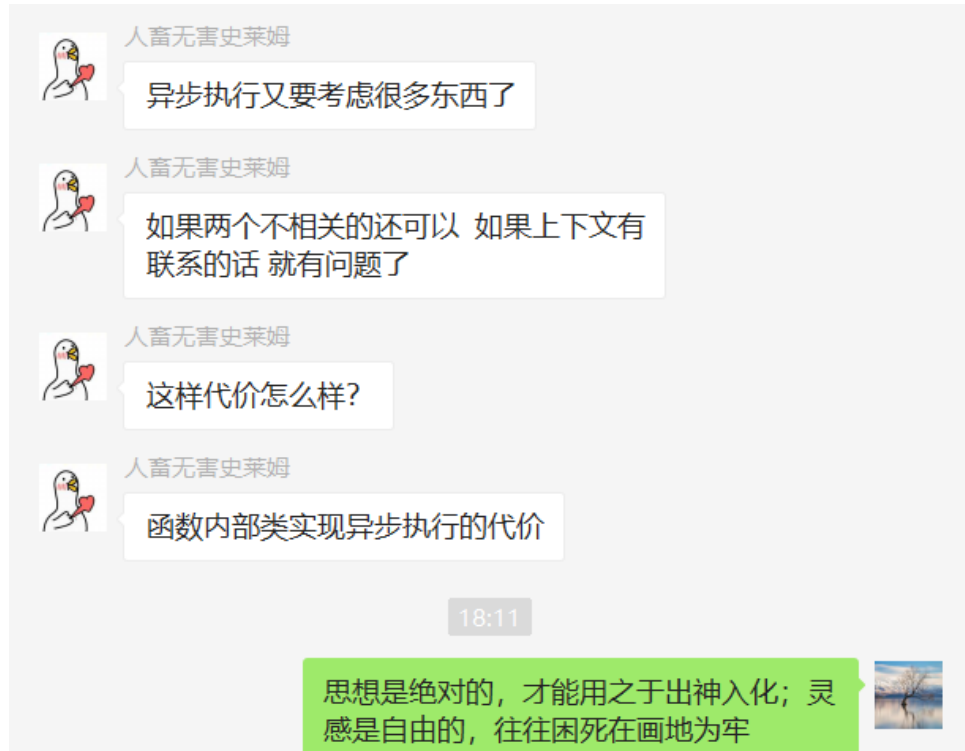
(3) 函数变成内部类之后，之前函数的函数体不变，而此时内部类提供 run 方法，去调用旧的函数体。

(4) 函数变成内部类之后，外部类增加线程池，外部类执行函数的过程变成了：将参数封装成内部类，然后扔到线程池，进行异步执行的过程。

后记：

有的人提了一堆的问题，对此类异议不再一一回复，统一回答为：

思想是绝对的，才能用之于出神入化；灵感是自由的，往往困死在画地为牢。



第六节：经常写技术文章的人未必代码境界高

在笔者看来，大多数喜欢写技术文章的人，水平都一般。这个结论的推导如下：

- (1) 人的精力是有限的，在这方面投入多了，在另外一方面必然少。
- (2) 写技术文章和写代码需要是思维方式是不一样的，甚至是矛盾的。一个人整天写些技术文章，必然导致思维方式已经不适应写代码，自然代码境界不会很高。

所以，笔者觉得：大多数喜欢写技术文章的人，水平都一般

笔者虽然也写一些技术文章，其实也适用用上述道理。刚开始技术水平有限，经常写些技术文章。后来技术提升之后，再也无往年之热情了。