

春节特别加餐讲王争：如何学习《设计模式之美》专栏



你好，我是王争。今天是春节，首先祝你在新的一年里，学业进步，工作顺利。放假期间，也不要忘记学习啊！

到此为止，咱们的《设计模式之美》快要更新过半了。很多小伙伴反馈的问题我都看到了。这里我也来讲一下，到底该如何学习《设计模式之美》专栏，才能做到事半功倍、有的放矢？

《设计模式之美》专栏相对于我的另一个专栏《数据结构与算法之美》来说，内容看似更容易，实际上却更难。为什么这么说呢？

从我写作的角度来说，《设计模式之美》要更加有难度。因为《数据结构与算法之美》的知识点都比较明确，一篇讲一个数据结构或算法。我只需要把原理、实现讲清楚，再举个实战的例子就基本大功告成了。

但是，《设计模式之美》中的知识点就没那么明确了。每个知识点到底该如何讲，并没有太多的参考。特别是专栏的前一部分，设计原则和思想，我在写专栏的时候参考了大量的书籍和资料。不瞒你说，大部分现有资料都讲得很浅，而且“互相借鉴”的痕迹也很明显。如果你之前对设计原则、思想、模式有所了解，或者看过挺多相关的书籍，应该能发现，我的专栏跟其他书籍的讲解还是有比较大的区别，这其中融入了我很多自己的思考和见解，这也是我的专栏与其他书籍区别最大的地方。

相对应的，从你学习的角度来说，《设计模式之美》也更有难度。它的难跟算法的难还不大一样。算法是原理难、实现难，最后才是应用难。而设计模式正好相反，原理、实现都挺简单的，基本上你一看就会觉得懂了，但是真正的能把它用到项目中，还是比较难的。对于很多人来说，你问他个具体的设计原则、思想、模式的原理和实现，他都能回答得头头是道，但是，在实际的项目开发中，写出来的代码质量还是很差。这种情况出现的原因还是相关的知识点都过于抽象，通俗点讲就是有点“假大空”，不够具体、不太能落地，导致理论和实践容易脱节。

所以，有些小伙伴总感觉《设计模式之美》学起来比《数据结构与算法之美》要简单，容易看懂，实际上这是种错觉。如果你

也有这种感觉，那你要警醒，因为你可能并没有真正读懂专栏的精髓。说了这么多，接下来我就讲讲，我为什么说你可能没有体会到文章的精髓，以及到底该如何来学习这个专栏。

1.建立完善的知识体系

你可能已经注意到，虽然这个专栏的名字叫《设计模式之美》，但专栏内容并不局限于此。实际上，整个专栏是围绕着教你如何编写高质量代码来设计的。

在整个专栏中，我试图给你建立完善的知识体系，所以，从内容上来讲，涵盖了编写高质量代码的方方面面，比如面向对象、设计原则、设计思想、编码规范、重构技巧、设计模式。虽然在有限的100篇文章、50万字内，我可能没法儿把每一个知识点都讲解得无比详细、全面，但我整理的这份知识框架，相当于给你指出了个深入研究的方向，你可以按图索骥，积累起来会更快。先建立整体的知识框架，再慢慢深入、各个攻破，这也是学习任何一门新技术、新知识最有效的方法。

所以，**这里给你布置一个小任务，闭上眼睛，想一想整个专栏都会讲哪些内容？已经讲过的设计原则和思想，都讲了哪些内容？你能想起来多少？**

2.建立代码质量意识

实际上，建立正确的技术认知和技术观，比单纯学技术、学知识点更重要。

我举个例子，专栏中讲到单元测试的时候，并没有讲跟某个测试框架相关的技术点。因为我觉得这些很容易通过看文档学习，而我也可能写得比文档更全面、更权威。实际上，通过那节课，我想要让你了解的是单元测试的好处，让你真正意识到单元测试的重要性，从心底认可这件事。还有，我在讲到重构的时候，也一直在强调要建立持续重构意识。整个专栏也都是在传递给你代码质量的重要性，让你意识到好代码和差代码的差别在哪里。

在学完专栏的时候，如果某些原则、思想、模式你记不大清了，那也没关系，只要你能在写代码的时候，能不由自主地去思考代码质量，有意识地去打磨代码，对代码质量有所追求了，那就说明你入门了，也就达到了我们专栏学习的目的。至于具体的知识点，随着时间的推移有所忘记，实际上并不碍事，多看几遍，多实践实践就好了。

这里，我也问你一个问题，**你可以想一想，你写代码或者读别人代码的时候，是否开始思考代码质量问题呢？如果还没有，那再给你布置一个任务，在今后的一个月里，写代码前、中、后，都思考一下代码的扩展性、可读性、可维护性、可测试性等代码质量问题，看看自己编写的代码是否符合这些质量要求，有没有需要继续优化重构的地方。**

3.主动学习而非被动学习

“师傅领进门，修行靠个人”，这句话说得特别好。同样听一个老师讲课，一个班级里面总有人会考90分，甚至满分，也会总会有人考不及格。差距在哪里呢？道理很简单，那就是你有没有用心学习。

类比到我们的专栏学习中，如果你只是走马观花地看一遍、跟听小说一样听一遍，收获肯定是甚微的，也就出现常说的“过不几天就忘了”的情况。因为这是一种被动学习方法，左耳朵进右耳朵出，脑子里怎么会留住东西呢？相反，如果你能学会主动学习、主动思考，遇到不会的、理解不了的知识点，自己主动去思考一下、查查资料，或者跟同事讨论一下，试着把专栏里的内容自己总结一下，认真思考每一个课后题，这样的学习效果要比被动学习强好几倍。

关于这一点，我也给你留个作业：**找一篇你觉得还没有透彻理解的文章，花上一天的时间，把里面的代码自己实现一遍，把文章的重点内容自己思考、整理，输出成文章。你可以看看，这样是否要比单纯看一遍收获更多呢？**

4.多读几遍更有收获

《设计模式之美》专栏的内容都不难理解，每篇文章仅有四五千字，十几分钟的音频，但是，多读几遍你就会发现，每读一遍都会有不同的收获。

如果你只是看一遍、听一遍，怎么能达到像我一样对知识点的理解程度呢？我自己在写专栏的时候，可是查阅了大量的文章和

资料。有的时候，一篇文章我要写好几天，这期间有长时间、高强度的阅读、思考和揣摩。你如果都不愿意花跟我一样多的时间，怎么能期望跟我有一样的水平呢？

一个人的认知和理解能力是受限於他的经历和经验的。如果你刚毕业不久，开发经验不多，看专栏的时候，难免会抓不住重点或者理解得不够透彻。那该怎么办呢？我的建议是，千万不要把学习专栏看成一蹴而就的事情，看完一遍就丢在一边了，而是要反反复复进行学习。看一遍理论之后，你可以在项目中尝试着实践一下，然后回过头来再看一遍，直到没有新的知识点可以汲取为止。这个过程可能需要持续很长时间，可能是1年、2年甚至是3年、5年，但只有这样，你才能积累出真正的能力、建立真正的竞争壁垒，而不只是学一些快餐知识、填补眼前的焦虑。

在《数据结构与算法之美》专栏中，我也说过，你要做一个长一点的学习计划。实际上，学习《设计模式之美》专栏也是如此，毕竟这两门课都是比较基础的，而且跟我们平时的开发比较相关，多花点时间在基础的知识上，收益要远大于学习很多花哨的新技术、新框架。

关于这一点，我再给你布置一个作业：**找专栏中一篇文章，反复读上10遍（当然是带着思考去读哈）。你看看是不是比只读一遍要理解得更透彻？是不是之前不能理解的地方自然就理解了呢？是不是每次读的时候都有新的收获？**

5.学会把代码写到极致

我经常说，写100段烂代码都不如写1段好代码，对代码能力的提高大。实际上，这个道理也可以应用到任何学习工作中，堆量只需要时间，但并不是每个人都能把事情做到极致，而能把一件事情做到极致的人，往往也能把其他很多事情做到极致。这也是为什么，很多人在某一行业做得很好，跨行去做另一个看似不相干的事情也同样能做得非常好。牛人往往都是能把一件事情做到极致的人。如果所有的事情都只能做到一般好，那你注定也只能做一个平凡的人。

关于这一点，我也布置一个作业给你：**找一段你觉得写得很烂的代码，花上一个礼拜的时间，反复思考如何优化，把它优化到足够好。**

总结

说了这么多，说实话，我知道这些道理很多人都知道。但是，能真正落实执行，并且执行到位的人不多。就光我说的这几个作业，估计能100%完成的也没有几个人。

你说人和人之间的差距在哪里？就在这里。

不要期望我的专栏有什么杀手锏可以教给你，不要期望看了我的专栏之后不费力气就能成为代码高手。还是那句话，师傅领进门，修行靠个人。我能做的就是尽量地将知识讲得通俗、透彻，把我的经验尽可能地传授给你，而这些只占1%，剩下99%都要靠你自己去努力。

如果说成为代码高手是万里长征，那我只能给你指明方向，告诉你如何去走。剩下的万里长征没人能替你去走，需要你一步一步、踏踏实实，自己去走完。

课堂讨论

最后，你可以在留言区讲一讲自己的**新年学习计划**。

如果有收获，欢迎收藏这篇文章，并把它分享给你的朋友。

精选留言



叫我天才好了

2020年学习计划：

- 1.跟着争哥的设计模式更新学习，并且回忆自己工作中那些还可以重构和完善的代码，虽然好多都是老东家的代码了，不过在新公司的代码一定要认真的、反复的保证完美。
- 2.把Spring课程跟着更新听下去，今年还要把买的两本Spring的书看完，跟着书练习一遍。
- 3.把MySQL搞定，书和课程。
- 4.2020年看完买来的关于专业知识的书，把大学必修的计算机基础的计算机网络、计算机组成原理、数据结构（争哥的课已经看完，准备看书+课程二刷+自己实现课程代码+LeetCode刷题）、操作系统都拿下。
- 5.不再焦虑，我只是一个刚工作1年多的萌新，焦虑太多也提升不了自己，还不如自律起来多学习多思考来得快。越努力，越幸运。

最后总结：加油吧，各位极客时间的小伙伴们！

2020-02-02 02:28



zliweijk

这个时势最好就是宅在家学习

2020-01-25 02:06



李小四

设计模式_春节特辑：

公司的一个实习生问我，你觉得技术好是什么意思：

我说：基础好！！

年度计划：自己的项目(app+server)上线~

问题1：闭上眼睛，想到的内容：

真的是盲想。。。

- 设计原则：

- 开闭原则

- 单一职责原则

- Kiss原则

- YGNNI

- DRY原则

- 。。。

- 几个重要的设计思想：

- 高内聚，松耦合

- 面向对象与面向过程的理念区别

- 继承与组合的取舍

- 实际的工作中，不要死板地遵守某一个(组)原则，要整体地取舍地考虑。

问题2：建立代码质量意识，

有思考，但之前没有从这么多的方面考虑(扩展性，可读性，可维护性，可测试性)，尤其是可测试性。

问题3：主动学习，找一篇还没有透彻理解的文章，理解后输出文章。

其实留言只是第一步，更好的方式一定是输出。我计划是“高内聚，松耦合”那一节。

问题4：多读几遍

10遍到是没有过，但专栏文章平均读3遍，可能是因为一遍真的看不懂吧。

问题5：找一段你觉得写得很烂的代码，花上一个礼拜的时间，反复思考如何优化，把它优化到足够好。

关于这个作业，我准备把项目中一个耿耿于怀的模块改了。

上面的这些作业，我希望自己是那个100%完成的人。

2020-01-29 09:41



唔多志

文章没有一句废话，都说到心坎里了。学习，永远都是自己的事。自己加油，中国也加油。

2020-01-26 00:49



javaadu

今年上半年自己的第一个目标就是完成设计模式之美的100天学习打卡活动。跟着老师学习的过程，再结合实际的工作项目，最近一个月感觉收获很大

2020-01-25 23:39



小可

鼠年大吉！

2020-01-25 09:43



Jxin

- 1.其实专栏名称改为《码出高效》更合适些（阿里的《码出高效》可以改名《面试基础知识点》）。
- 2.19年给自己定的目标恰巧就是写出高质量代码，虽然现阶段造不了飞机，但就算是做茶壶也要做展览台的艺术品，而非街头摊子的劣质品。
- 3.栏主这个专栏，涵盖好几本书的内容。个中的文字描述令人惊叹，自愧不如。19年底也算是借着这个专栏把自己关于高质量代码的知识体系梳理了一遍。万分感谢。
- 4.就个人拙见,关于码出高效，技术相关其实还少了运行监控这块的东西，落地到代码，包括各种日志的输出原则，系统和业务健康指标的选择等等。非技术的，还有产品思维和精益思想需要学习，我们的目标是更高效的产出业务价值，这本身就很难逃过对业务背景必要的学习。而建模抽象，需求优先级排序以及需求设计都需要产品知识的支撑，技术,产品对于业务开发本就是一体两面的东西。

新年计划：

- 1.决定公司内部分享《码出高效》一整套的课程（文档，ppt，视频，github项目），过年都在筹备。面向对象和设计原则这两块就以栏主的内容为基础来写，毕竟目前我没本事写得更好也没必要去另起炉灶（会著名出处的）。

2020-01-27 15:00



郭凯强

新年准备把数据结构和设计模式这两个专栏分别做好对应的笔记。能用自己的话写出来，才算真正的理解

2020-02-04 10:08



batman

缺的就是这1%的方向，剩下 99% 都要靠你自己去努力，老师的肺腑之言记住了！

2020-01-30 21:51



志振爱

学习完之前累积的极客时间课程

每周做一个leetcode

寻找新的方向，新的起点

2020-01-30 19:45



L

新年计划：把基础专栏学好，学完了再学进阶的

2020-01-28 20:43



下雨天

对于设计模式之美，一步一个脚印来，每一篇都要做笔记有输出，学过的内容，忘记的内容，回过头来重复重复再重复！杜绝学过就是学到，学一遍就万事大吉的想法！

2020-01-27 09:45



xavier

新的一年，先打好基础，修炼好内功。

2020-01-26 21:20



一步

文章的方法，不仅是学习设计模式的方法，有些还是如何学习的方法

2020-01-26 20:50



PHP是世界上最好的需要



以前写代码，蒙头写，现在写代码，立马能警醒自己，是否开闭，是否可复用，是否有可读性，维护性，命名是否合适，都会有一个比较有自信的理论基础，确实收货不少，今年再接再厉

2020-01-26 19:16



衛嚼爺

2020计划：好好认真的学习买的专栏，特别是基础性的专栏，把基础打好。然后再考虑高阶的

2020-01-26 13:41



Eden Ma

闭上眼睛发现该复习复习了哈哈

2020-01-26 11:29



AaronYu

新年计划：把专栏跟下来，有一个大概印象，以后写代码时能有个基本概念和要求，再持续精进，逐步掌握。

2020-01-25 23:00



Frank

纸上得来终觉浅绝知此事要躬行。

新年计划：跟着专栏一步一个脚印学习，遵循输入（阅读）-处理（思考，写代码）-输出（写留言，记笔记等）的思维模式稳扎稳打。

2020-01-25 22:39