

# 开篇词 | 一对一的设计与编码集训，让你告别没有成长的烂代码！

王争 · 设计模式之美



你好，我是王争，是“[🔗 数据结构与算法之美](#)”专栏的作者。“数据结构与算法之美”专栏在今年 2 月底全部更新完毕。时隔 8 个月，我又给你带来了一个新的专栏“设计模式之美”。如果说“数据结构与算法之美”是教你写出高效的代码，那这个设计模式专栏就是教你写出高质量的代码。

## 程序员的看家本领你得练好

研究生毕业后我就加入了 Google，至今我还清晰地记得，我第一次提交代码的时候，短短的 100 多行代码，被同事 review 出了 n 多问题，来来回回改了不下十几个版本才提交上去。我当时有很大的逆反心理，觉得有必要浪费这么多时间在如此细节的编码上吗？只要代码能用、能解决问题不就够了吗？

工作一段时间之后，我才发现自己当时的想法有多幼稚。

写代码可以说是程序员天天要干的事情，要是代码都写不好，最基本的看家本领都练不好，成天堆砌烂代码，写代码还有啥意思呢？那还干啥程序员啊！写出“能用”代码的人比比皆是，但是，并不是每个人都能写出“好用”的代码。只会写能用的代码，我们永远成长不成大牛，成长不成最优秀的那批人。

后来我熟练掌握了各种编写高质量代码的技巧、方法和理论，我发现，实际上，写烂代码和好代码花费的时间是差不多的。当你把写高质量代码培养成一种开发习惯之后，在你在编写代码的时候，自然就有一种代码质量意识，自然而然就可以写出不错的代码。即便在我离开 Google 加入其他公司之后，项目的代码质量因为各种原因有所妥协，但我起码知道什么样的代码是高质量代码，丝毫不影响我具备写出高质量代码的能力。

我相信，很多工程师都很重视代码质量，毕竟谁也不想写被人吐槽的烂代码。但是，就我的了解来看，毫不夸张地讲，很多工程师，甚至一些 BAT 的员工，代码都写得惨不忍睹。一方面，在目前这种快糙猛的开发环境下，很多工程师并没有太多时间去思考如何写高质量代码；另一方面，在烂代码的熏陶下，在没有人指导的环境里，很多工程师也搞不大清楚高质量代码到底长什么样。

这就导致很多工程师写了多年代码，代码功力一点都没长进，编写的代码仍然只是能用即可，能运行就好。平日的工作就是修修补补、抄抄改改，一直在做重复劳动，能力也一直停留在“会干活”的层面，就像高速公路上的收银员，只能算是一个“熟练工”。

## 一个人闷头看书效果并不好

当然，也有一些比较上进的工程师，会去找设计模式、编码规范、重构等类型的书籍去看，学习如何编写高质量的代码。实际上，我也买了很多这类的书籍来看，从这些经典的书籍中，我也学到了很多编程技巧和提高代码质量的方法。

不过，这些书籍都有一个特点，那就是比较偏重理论讲解，喜欢拿猫、狗之类生活中的例子来举例。当然，这样的例子也有优点，那就是能在简短的时间和篇幅内，很好地帮你理解原理。但同时也存在一个严重的问题，那就是过于脱离真实的软件开发。而且例子本身没有难度，你一看就觉得懂了，但是看完之后，可能还是不清楚如何将理论落地到实际的项目编码中。

比如，我们都知道著名的 KISS 原则（Keep It Simple and Stupid）。这个原则理解起来很简单，一看貌似就懂了，那我问你，怎样的代码才算是足够简单呢？怎样才算不够简单需要优化呢？估计很多人都回答不上来，因为大部分书籍都没有讲清楚。

除此之外，一个人自己闷头看书，在很多时候效果并不好。一方面，每个人的理解能力是不一样的。对于同一本书，不同理解能力的人看完之后收获也是不一样的。跟着有经验的老师学比闷头自己看书要更高效、收获更多、成长更快。另一方面，编码本身就是一门实践课，光闷头看书本理论肯定是不够的，更重要的是在实践中学习如何应用这些理论。

## 一对一手把手指导才最有效

从我的经验来看，我觉得最有效、最快速提高编码能力的方法就是，找一个比你资深的工程师，一对一、手把手地指导你写代码。你提交代码，他来指出你的问题，你再优化，这样一来一往，要不了多久，你就会发现，自己的代码能力突飞猛进。

但是，理想很丰满，现实很骨感。且不说能不能找到这样有资格指导你的人，即便能找到，他愿不愿意、有没有时间来手把手指导你，还是另外一回事。而我比较幸运，在毕业之后就加入了 Google，得到了顶尖工程师的指导，一对一地给我 review 代码，手把手地指导我如何优化代码。正因如此，在 Google 的那段时间也成为了我编码能力提高最快的一段时间。

所以，在设计专栏的初期，我就在想，如果我能模拟这样一个一对一、手把手、就真实项目代码讲解的场景，是不是就能让专栏有别于千篇一律的书籍，从而能真正提高你的代码能力呢？基于这样一个想法，我们接下来就来看，我是如何设计整个专栏内容的。

## 我是如何设计这个专栏的？

### 100 多篇文章、50 万字、2 万多行代码

整个专栏的文章总共有 100 多篇，每篇平均下来在 5000 字左右，所以你总共需要学习 50 万字。为什么篇幅会这么多？这是因为，我想一次性把跟编写高质量代码相关的所有知识，都系统、全面地讲清楚，一次性给你讲透彻。你看完我这一个专栏，就能搞清楚所有跟写高质量代码相关的知识点。

除此之外，为了避免脱离代码，空洞地讲理论，专栏每篇文章平均大约有 200 多行代码，整个专栏累计有 2 万多行代码。而且，这些代码都来自我这十几年积累的真实项目，而非编造出来的阿猫阿狗、停车场、餐厅之类的没有太多实际意义的代码。

我个人写 Java 代码比较多，所以，专栏中的代码我是用 Java 语言实现的。不过，在设计专栏的时候，我已经考虑到其他不熟悉 Java 语言的小伙伴了。我力争做到，实际上我觉得也已经做到，一方面，专栏内容的讲解并不与具体的编程语言挂勾；另一方面，我只用最基本、最简单的 Java 语法，只要你有一些编程基础，不管你熟悉的是哪种编程语言，都可以看懂我写的代码，理解代码背后的设计思想。

所以，如果你熟悉的不是 Java，而是 C++、C#、PHP、Python、Go、Ruby、JavaScript 等其他编程语言，不要担心，这完全不会影响你学习这个专栏。

## 200 多个真实的项目实战代码案例剖析

实际上，大部分设计原则、设计思想、设计模式理解起来都不难，难的是如何将这些理论灵活恰当地应用到实际的开发中，而不是教条主义和盲目滥用。而要想正确、得当地应用理论知识，光看书是不够的，我们要在实战中去亲身感受、体会这些理论该如何应用。但是，我们平时的开发更多的是基于已有的框架，照着别人的流程，扩展新的功能模块。所以，在工作中就可能没有那么多足够复杂、足够有难度的开发场景，让我们有机会去实践这些理论。

基于这种考虑，我结合自己过去工作中积累的项目经验，为整个专栏精心设计了 200 多个真实的项目实战代码案例。几乎每节课、每个知识点都会结合一个完整的开发案例来讲解。我们知道，刷算法最出名的就属 LeetCode 了。类比一下的话，我这 200 多个开发案例足以称得上是一个锻炼编写高质量代码的“LeetCode”了。跟着我一起把这 200 多个开发案例练下来，不愁代码能力提高不了！

## 8 个月、240 天一对一手把手编程指导

前面我有提到，提高自己编码能力，最有效的手段就是让有经验的前辈一对一、手把手地指导。我经历过 Google 最严格的 Code Review，也被最顶尖的工程师指导过，也见识过最优秀的代码长什么样子，而且自己平时也比较重视代码质量，所以，我觉得我还是有挺多写代码方面的心得体会可以分享给你的。

而且，看过“数据结构与算法之美”专栏的小伙伴都知道，我这个人最大的特点就是逻辑清晰，能抓住要点把问题讲透彻。所以，对于专栏中的理论解读，我并不会照搬那些经典的大部头书籍，而是更多地融入我自己的思考和理解。或许我的解读会跟大师的不完全一样，但我都可以自成一体，并且告诉你如何落地指导具体的开发、编码，而不是很鸡汤地浮在表面来讲，让你听的时候感觉好像都懂了，感觉思想好像得到了升华，而合上书就又忘得一干二净，更别提应用到项目中了。

当然，除了理论解读之外，对于每一种设计原则、思想或者模式，我还会讲一些更深入、更本质、更有用的东西，比如，“为什么要有这种设计原则、思想或者模式？它能解决什么编程问题？有哪些应用场景？又该如何权衡、恰当地在项目中应用？”等等。

在讲解每个实战案例的时候，我会从最原始的问题代码讲起，然后告诉你缺陷在哪里，如何通过设计原则、思想、模式来优化，让你知其然知其所以然，了解背后设计的来龙去脉。我觉得，这比单纯只告诉最终的设计结果要重要得多。我也希望，你不光是被动地听我讲，而是能跟着我的节奏主动地去思考，这对你的逻辑思维训练也会很有帮助。毕竟，对于成天跟程序逻辑打交道的程序员来说，逻辑思维能力是一项非常重要的能力。我们平时要多多加强这方面的锻炼。

除此之外，专栏内容的讲解，我也刻意模拟这种一对一指导的感觉。虽然我没法真正坐在你的面前，跟你一块讨论原理、分析如何优化代码，但我力争让你在看或者听专栏的时候，就好像我在你的面前，跟你一块讨论问题一样。整个专栏有 100 多篇文章，一周更新 3 篇，需要持续更新 8 个多月、240 多天。这就相当于我手把手指导你 8 个月写代码。我希望通过这 8 个月的指导，能把我十几年的积累统统传授给你。

## 100 多个有深度的课堂讨论、头脑风暴

我们知道，设计问题本来就没有标准答案。所以，不要轻信一家之言，更不要迷信我的专栏，也不要盲目追从任何一位专家、大师的说法。主动思考，积极讨论，比单纯地被动接受，学习效果要好十倍、百倍。

所以，在每节课的最后，我都设计了 1~2 道课堂讨论题，这些题目有些是代码设计相关的开放性问题，有些是具体的代码优化问题。看过我“数据结构与算法之美”专栏的小伙伴们都知

道，专栏的思考题和同学们的留言，有的时候比专栏文章本身还要精彩。所以，对于这个专栏，我也希望你能积极参与，发表自己的观点。

如果你在学习完每节课程之后，都能花 5 分钟、10 分钟的时间去思考一下课堂讨论问题。这不仅对课程知识的学习很有帮助，还能锻炼你的思考能力。你要知道，独立思考能力对一个人来说是多么重要啊！

## 让我们一起见证成长

我个人觉得，“数据结构与算法之美”和“设计模式之美”是两门跟我们每天的编程开发，直接关系非常大的课程，是两门奠定一个工程师最基本的代码能力的课程。它们一个教你如何写高效代码，一个教你如何写高质量代码。弄懂了这两门课的内容，在今后的开发中，你就不需要担心写出被人吐槽的烂代码了，甚至还能让你的代码成为职场发展的一个加分项和闪光点。

前面讲到，专栏要持续更新 8 个月的时间，你可能会觉得时间好长。实际上，只要跟着专栏更新的节奏，每周认真学好 3 节课，稳扎稳打，8 个月一眨眼就过去了。而经过这 8 个月的刻意编码训练，我相信，你会发现自己因为这样的坚持学习成长了很多，不仅代码能力会提高几个档次，或许还能养成一种好的学习习惯。

最后，今天是开篇，你可以做个自我介绍，立个学习 flag。当然，你也可以留下你对专栏的期待和建议。

说起来连续 8 个月坚持学习，不是件容易事，而和志同道合的小伙伴一起，更容易找到学习的乐趣。为此，我在“极客时间小程序”发起了“8 个月，攻克设计模式”的学习打卡活动。我可以向你保证，这 8 个月只要你跟着我的节奏，踏踏实实走下来，你的代码能力一定能有一个质的飞跃。

欢迎加入，让我们一起见证成长！

### AI智能总结

作者王争在新专栏“设计模式之美”中分享了如何写出高质量的代码。他强调写好代码是程序员的基本技能，而只会写“能用”代码无法成为顶尖工程师。专栏包含100多篇文章，总计50万字，涵盖了编写高质量代码的所有知识点，并附有2万多行真实项目代码。通过一对一的指导来快速提高编码能力，类似于他在Google受到的指导。专栏内容的讲解模拟一对一指导的感觉，力求让读者在学习时感觉像在接受作者的指导。文章还提到了200多个真实的项目实战代码案例，以及8个月、240天的一对一手把手编程指导，让读者能够系统全面地学习

写高质量代码的知识，从而提升自己的编码能力。文章还强调了参与课堂讨论的重要性，以及通过持续学习来见证成长。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

## 全部留言 (685)

最新 精选



符明明

2019-11-07

我自我介绍一下，985本科计算机系第一名，三年国奖，加拿大全额奖学金满绩点硕士毕业，编程大赛第一名，数学天才。我立的flag，进去美国微软西雅图总部工作，把创业公司搞上市，已经完成一半。

作者回复: 😂

共 34 条评论 >

👍 209



爱吃锅巴的沐泡

2019-11-04

老师会在加餐时间讲一讲uml的知识点吗，感觉设计模式少不了uml

作者回复: 感觉uml没啥用，专栏后面会解释这一点的。

共 8 条评论 >

👍 22



Nova - 斌神

2019-11-09

工作七年的BAT老鸟工程师也来跟争哥一起学习写代码，把落下来的看家本领捡起来

作者回复: 互相学习~

共 2 条评论 >

👍 14



上下求索

2020-07-29

5年的菜鸟，不知道学这些有没有用呢？我不知道未来走那条路，我很迷茫，目前没去过一家互联网，做的项目也简单，今天是进外包的第一天，心里沉到了谷底。我试着每天学一篇吧。。。

作者回复: 5年还好，路还很长呢~你抓紧看，先搜一遍，然后再细看，看看能不能打开你的思路

共 2 条评论 >

👍 4



张嘴儿

2019-11-18

课程什么时候能更新完呀

作者回复: 明天6月



👍 4



斜杠ing...

2019-11-14

为什么都看《大话设计模式》，那本书我觉得很不好啊，里面的例子很勉强，不如《设计模式之禅》啊

作者回复: 大话 写的有点啰嗦



👍 4



搅局

2019-11-06

世间所有的相遇都是久别重逢。如果你在十几年前关注了Mac，七八年前在博客园关注了老池，那么大概率在2019年的今天遇到这个精品课程。#首日打卡#

作者回复: 也可以直接关注我wx公号 小争哥 😂



👍 3



100824

2020-07-31

字节程序员也来充充电，flag 每天学一讲



作者回复: 大佬你好, 能内推吗💎💎💎💎💎💎

共 2 条评论>



2



陆君辉

2020-11-29

立下flag, 必须完成所有课程, 课后练习要完成

作者回复: 💎💎💎💎💎💎 加油



777

2020-11-25

来学习啦!

作者回复: 加油

