



下载APP



## 结束语 | 写代码是一件可以一生精进的事

2021-02-09 郑晔

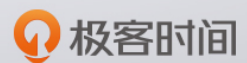
代码之丑

[进入课程 >](#)**郑晔**

推文科技技术 VP、前火币网首席架构师

你好，我是郑晔。

写代码是一件可以持续一生的事情，但前提条件是，你得找到自己的热爱，并建立起自己的正向反馈。

**讲述：郑晔**

时长 09:44 大小 8.93M



你好，我是郑晔。

春节将至，祝你新春快乐！我们的专栏到这里也正好要更新完结了，在结束语这一讲，我想和你聊聊程序员精进的话题。

创作《[10x 程序员工作法](#)》之初，我曾经定下了“写下 100 篇”的宏伟目标。在第三个专栏结束的时候，这个当年许下的宏伟目标终于实现了。

如果为这 100 篇的内容找一个共同的主题，那就是程序员精进之路。



### 程序员精进之路

在很多人心目中，程序员是一个辛苦的职业，一方面，各种新东西层出不穷，程序员们要努力追随，另一方面，业务飞速发展，我们唯有积极应对。那么，是什么支撑你在这个富有挑战的行业里坚持前行呢？

于我而言，这个问题的答案是，热爱。

在我的心目中，编程是一项有趣的智力活动，从最初解决一个特定的小问题，到现在创造一个方案去解决一个系统的问题，无不需要费尽心力去探寻一个好的解决方案。时至今日，即便我写程序已经二十多年了，但每次程序运行通过时，我心里依然还是有一些小激动，因为支撑程序运行的每行代码里都有自己的思考在里面。

正是每次一点点积累起来的成就感，激励着我不断去探索更好的做法。一开始，为了解决特定的问题，我四处搜集着各种编程技巧。当我理解了基本功的重要性后，就去拼命地补齐各种基础知识，构建起一个知识体系。随着开源软件运动的兴起，我知道了，原来有各种工具和程序库可以简化自己的工作。因为见识过别人的运指如飞，我曾专门练习了各种快捷键和命令行。

当我已经能够很好地解决自己面对的各种功能问题时，我开始抬起头，有了更大的视野。

我学习了各种软件设计的知识，让自己的代码不仅仅是为了今天，也能够面对未来。我学习了各种程序设计语言，看到了隐藏在语言背后的编程范式和思考习惯。我学习了各种软件开发的最佳实践，懂得了怎样让一群人更好地协同。

我做的所有努力，都是为了更好地写代码。

有了对于软件开发更多的思考，回过头再来写代码时，我就能看到更多的维度，能意识到自己在写的代码对他人和未来的影响，这时自然会尽力把自己的代码写得更整洁。

时至今日，如果你问我，对自己写的代码满意吗？我的答案还是不满意。**写代码是一门手艺，需要不断地打磨。**一方面，坚持写代码，保持自己对于代码的体感；另一方面，保持对于代码的敏感度，不断思考对于代码的改进，寻找更好的写法。

经过一段时间，我总会发现代码中让我不满意的地方，这会成为新的驱动力，让我进一步扩充自己的知识，把新的理解注入到代码之中。每次拓展知识边界，与之相伴的都是极大

的智力愉悦。也正是这种智力上的快感，让我得到了进一步前进的动力。一个正向反馈的循环就是这样逐步推进，让我在写了二十多年代码之后，依然乐此不疲。

我在《[🔗 软件设计之美](#)》中讲过，一个好的设计是在一个“小内核”上构建起来，然后，逐步添加更多模型。我们的知识拓展过程也是如此。我的“小内核”就是编写代码这件事，所有一切知识的拓展都是围绕这个内核展开的。

**写代码是一件可以持续一生的事情，但前提条件是，找到自己的热爱，建立起自己的正向反馈。**坚持写代码，发掘代码中值得改进的地方，不断拓展自己的知识边界，寻找更好的代码写法，这就是最朴素的程序员精进之路。

## 代码的敏感度

对于一个不断精进的程序员而言，发掘代码中值得改进的地方，需要对代码有细致入微的敏感度，这样才能体察代码间细微的差别。

我给你举个例子：注释。

代码该不该写注释呢？在一些人看来，没有注释的代码不值得写，这甚至成了一些程序员的宗教信仰。如果你问他们为什么要写注释，他们的回答多半是“让程序更加容易理解”。

但有另外一群人则将注释视为坏味道，他们会说，为什么不把代码写得更清楚，让代码不需要注释呢？被逼到角落的“注释程序员”依然不会束手就擒，他们不会承认自己不能把代码写清楚，而会说，有些代码必须要有注释才能解释清楚，比如一些算法。

好了，双方的主要观点陈述完毕。你怎么看待注释呢？

早在 1984 年，《计算机程序设计艺术》的作者 Donald Knuth 就给出了一个回答，他提出了“[🔗 文学编程 \(Literate Programming\)](#)”的概念，其核心要义就是要将程序写得像用自然语言进行表达一样顺畅。虽然作为一种编程范式，它并没有流行起来，但它背后蕴含的思想却影响了很多，也给我们提出了更高的技术追求。

一个好的程序应该像一篇优美的文章，读起来自然流畅，二者背后有诸多相通之处，**你会看到，许多优秀程序员都有着优秀的表达能力。**所以，回到写代码本身，把程序本身写得

更清楚直白才应该是我们的追求。关于如何把代码写好，我在这个专栏已经讲了很多了。

具体“注释”这件事上，我的观点是，“注释”有其价值，但不应该是主力。我们没见过哪篇文章是要求把注释作为主旨的，同样，过于强调注释，无异于本末倒置。写代码首先应该是把代码本身写好，至于那些确实无法用代码陈述清楚的部分，我们再考虑用注释。

所以，我赞同把“注释”当做坏味道的提示，**先竭力把代码写到不需要用注释，而把注释当作最后的选择**。确实有一些特定的处理需要注释，无论是一个精巧的算法，还是一个特殊的技巧。用这个标准要求自己，你会发现，大多数代码其实不需要注释，因为它们太普通了。

你看到了，即便像注释这么简单的东西，写与不写，背后都有着可以探究的各种细节。诚如我在前面所说，写代码是一门手艺，需要不断地打磨。唯有无限界地拓展自我，才可能对代码有细致入微地把握。

你发现了，打磨手艺，锤炼自己对于代码的敏感度，坏味道是一个不错的出发点。这也是我写这个专栏的初衷，帮你从识别出那些你曾视而不见的“坏味道”，提升你对代码的敏感度。

在这个专栏中，我给出的就是全部的坏味道吗？显然不是。只要拿出《[重构](#)》对照一下，你就会发现，坏味道还有许多，比如，霰弹式修改和发散式变化。我没有拿出来讲，不是它们不重要，而是它们不像我在这个专栏中罗列的这些坏味道那样，有非常直观的表现。

比如，霰弹式修改说的是一次变化要在很多类的内部做修改，但能否察觉出自己改了很多类，这就依赖于每个人的敏感度了。

同样，发散式变化说的是，不同的变化都会改到同样的模块上。发现这种坏味道，需要你意识到，对同一个模块的修改是由于不同的原因造成的，这对于敏感度的要求就更高了。

无论如何，“知道”有哪些坏味道是第一步的。我建议你在学习了本专栏之后，花上一点时间，通读一下《重构》的第三章“代码的坏味道”，在开篇词中我就提到过这件事，但与那时不同的是，现在你已经通关了我们这个专栏。

我在专栏里讲的所有这一切，一方面，让你对一些代码的坏味道有直观的认识；另一方面，也是更重要的，对于这些坏味道的分析，是为了帮你看到代码里的细微之处，帮助你提升对于代码的敏感度。有了不同的敏感度，再去通读“代码的坏味道”，你会有不一样的收获。

这些道理都是知易行难，今天的课程就告一段落了，但是你的精进之路并未停止。

有了“坏味道”的基础之后，接下来最重要的是，你要在实际的工作中反复地锤炼自己的编程手艺，用这些坏味道作为尺子，衡量自己的代码，不断地找到代码更好的写法。这个专栏以及它的两个“兄弟”，帮你开启了程序员的精进之路，但这条路，总归还是要自己去走！

这次的《代码之丑》的旅程就暂告一段落吧，如果以后有机会，我会再来与你分享我对软件开发的理解。

专栏结束了，你也可以在留言区提问，我看到还是会回复的。也欢迎你点击下面图片，为我的《代码之丑》评分、提建议。期待你的反馈，再见！



郑晔

推文科技技术 VP、前火币网首席架构师

感谢一起走过的这段时间，非常想听听你对我和这门课程的反馈与建议。在 2021 年 2 月 22 日前提交，将有机会获得



原创 | 极简时尚双肩包

价值 **¥129**

或



极客时间课程阅码

价值 **¥99**

填写问卷 

[提建议](#)

12.12 大促

# 每日一课 VIP 年卡

10分钟，解决你的技术难题

**¥159/年** ~~¥365/年~~每日一课  
VIP 年卡**仅3天，【点击】图片，立即抢购>>>**

© 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。

[上一篇](#) 17 | 课前作业点评：发现“你”代码里的坏味道[下一篇](#) 结课测试 | 这些代码坏味道的知识你都掌握了吗？

## 精选留言 (9)

[写留言](#)**穿越时间与空间的旅行...**

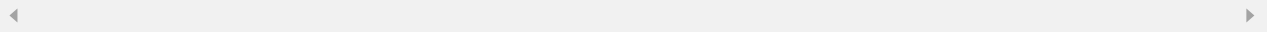
2021-02-09

感谢老师的分享，由于最近项目重构，学了老师的课感觉受益非浅。中间又去听了10x程序员和软件设计之美。希望能早日看到老师的新课程。

[展开](#)



作者回复: 三合一味道最好



qinsi

2021-02-09

个人感想:

\* 郑老师从业二十多年依旧在打磨编程手艺，瑞思拜！

\* 依稀记得RUP刚出来的时候，就有人认为今后程序员会被淘汰，因为你甚至可以画画图...  
展开 ∨

作者回复: 手艺的重要性，取决于你给自己的定位。

从行业整体的角度上，写一个 App 或是网站的速度，确实是越来越快了。这是建立在我们现在有了大量的基础设施上的，有人在最初设计好了架构，其它的程序员都是在这个架构基础上进行构建。这就在建筑行业，有人是设计师，有人是负责施工，你想扮演什么角色呢？

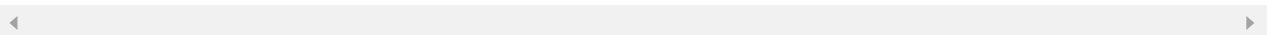
建筑的设计师，他们其实也是艺术家，但其实他们也是手艺人，市面的建筑工人可以找到很多，但建筑设计师却很难找到一大把。

在程序员这个行业里，优秀的架构师和普通的代码编写者之间并没有一个清晰的界限，他们都在写代码，但他们写的代码重要性是截然不同的，当然，他们的身价也是截然不同的。

你去问任何一个架构师，他绝对不屑于只完成功能，他肯定有更高的追求，无论是让系统的架构更有可扩展性，还是让系统承载更大的容量，抑或是让系统变得更快。这些都是一种不断打磨的手艺，需要不断地锤炼。

当你一不小心走到一个公司或是一个领域的最前列，能够给予你指导的，不再是别人已有的解决方案，而是你的品位，而品位的锤炼就是在日常中不断打磨出来的。

回到你的问题上，你可以选择永远做一名普通的编码者，在别人画好的圈子里面跳舞，但也没有资格抱怨，世界对你的不公，为什么程序员是有天花板的。你也可以让自己不断地提升，去追求更好的手艺，成为那个让别人可以依赖的人，去解决更多好的问题。



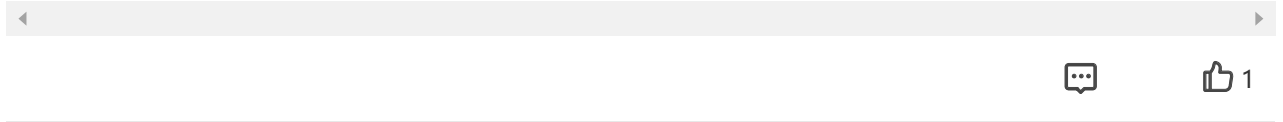
Jxin

2021-02-09

- 1.结束得太快了。100章结束了，下一站做啥？期待下个专栏。
- 2.感谢郑老师的分享，我们下次再见。

展开 ∨

作者回复: 可能会考虑在这个专栏做个加餐，分享一些写代码的思考。

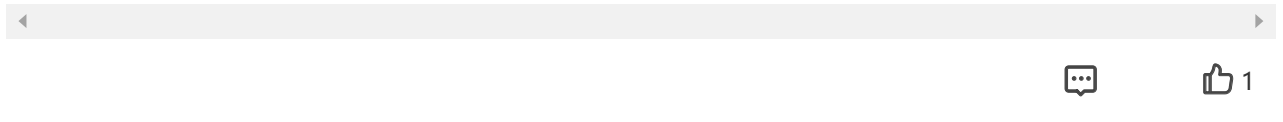
**Edison Zhou**

2021-02-09

正在再次复习重构的第三章，呼呼。

展开 ∨

作者回复: 完美匹配

**webmin**

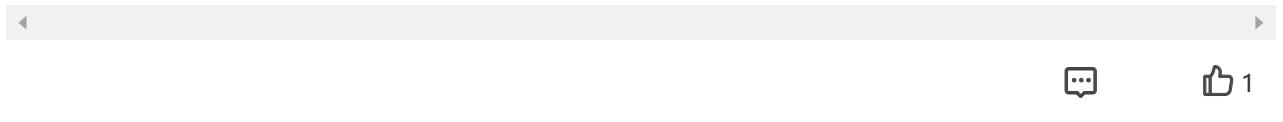
2021-02-09

关于注释最近我也冒出了困惑，事由是这样的新来了一位同事他写注释很勤奋，几乎达到一行代码一行注释，当时隐隐觉得不对劲，Review他的代码后，发现基上是业务怎么要求他就怎么翻译为代码，缺少聚合和封装，因为大家不是一个项目组，且我也不负责考核他，所以就没有去做进一步的事。

...

展开 ∨

作者回复: 每每写专栏，于我自己而言，就是一遍痛苦的梳理过程。你们看到的稿子都是几经打磨之后的产物。

**布凡**

2021-02-09

上班最后一天，学完了老师的代码之丑，也愿自己在写代码的路上越来越精进，不断超越自己，也祝老师和同学们春节快乐(\*^\_^\*)

展开 ∨

作者回复: 加油，努力不能停



**捞鱼的搬砖奇**

2021-02-11

当年听你的10x 的做了个一份xmind <https://gitee.com/meiyoudaima/xmind/tree/master/10X>

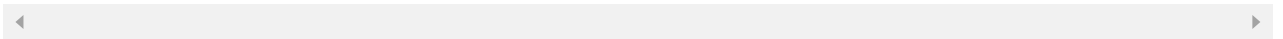
现在这门课收尾了，准备整理下，再做一次记录

谢谢老师，期待下一门课程。

祝老师新年快乐。

展开 ▾

作者回复: 欢迎分享

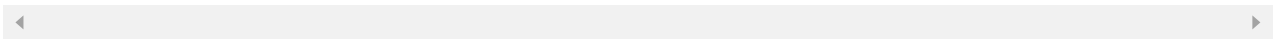
**Geek\_3b1096**

2021-02-11

老师新年快乐！

展开 ▾

作者回复: 过年好！

**Sinvi**

2021-02-09

把写代码也当成艺术

展开 ▾

