01 | 为什么说每个程序员都要尽早地学习并掌握设计模式相关知识?

王争 · 设计模式之美

我相信，很多程序员都已经意识到基础知识的重要性，觉得要夯实基础，才能走得更远，但同时对于如何将基础知识转化成开发“生产力”仍然有些疑惑。所以，你可能看了很多基础的书籍，比如操作系统、组成原理、编译原理等，但还是觉得很迷茫，觉得在开发中用不上，起码在平时的 CRUD 业务开发中用不上。实际上，这些基础的知识确实很难直接转化成开发“生产力”。但是，它能潜移默化地、间接地提高你对技术的理解。

不过，我觉得，设计模式和操作系统、组成原理、编译原理等这些基础学科是不一样的。它虽然也算是一门基础知识，但是它和数据结构、算法更像是一道儿的，相比那些更加基础的学科，设计模式能更直接地提高你的开发能力。我在开篇词里也说了，如果说数据结构和算法是教你如何写出高效代码，那设计模式讲的是如何写出可扩展、可读、可维护的高质量代码，所以，它们跟平时的编码会有直接的关系，也会直接影响到你的开发能力。

不过，你可能还是会觉得设计模式是把屠龙刀，看起来很厉害，但平时的开发根本用不上。基于这种观点，接下来，我们就具体地聊一聊，我们为什么要学习设计模式?

1. 应对面试中的设计模式相关问题

学习设计模式和算法一样，最功利、最直接的目的，可能就是应对面试了。

不管你是前端工程师、后端工程师，还是全栈工程师，在求职面试中，设计模式问题是被问得频率比较高的一类问题。特别是一些像 BAT、TMD 这样的大公司，比较重视候选人的基本功，经常会拿算法、设计模式之类的问题来考察候选人。

所以，我在求职面试的时候，都会提前准备、温习一遍设计模式。尽管并不是每次面试都会被问到，但一旦被问到，如果回答得不好，就是一个败笔，这场面试基本上也就凉凉了。所以，为了保证万无一失，摆脱一旦被问到答不出来的窘境，对于设计模式这种大概率被问到的问题，我都会未雨绸缪，提前准备一下。

当然，我并不是临时抱佛脚。我平时就比较重视设计模式相关知识的积累，所以底子比较好，只需要在每次面试前花很短的时间，重新温习一下，便可以自信满满地去面试，而不是心里老是担心被问到，影响正常的面试发挥。

所以，如果你也不想让设计模式相关问题成为你面试中的短板，那跟着我把专栏中的知识点都搞清楚，以后面试再遇到设计模式相关的问题，就不会惧怕了，甚至还会成为你面试中的亮点。

2. 告别写被人吐槽的烂代码

我们经常说，“Talk is cheap，show me the code。”实际上，代码能力是一个程序员最基础的能力，是基本功，是展示一个程序员基础素养的最直接的衡量标准。你写的代码，实际上就是你名片。

尽管我已经工作近十年，但我一直没有脱离编码一线，现在每天也都在坚持写代码、review 指导同事写代码、重构遗留系统的烂代码。这些年的工作经历中，我见过太多的烂代码，比如命名不规范、类设计不合理、分层不清晰、没有模块化概念、代码结构混乱、高度耦合等等。这样的代码维护起来非常费劲，添加或者修改一个功能，常常会牵一发而动全身，让你无从下手，恨不得将全部的代码删掉重写！

当然，在这些年的工作经历中，我也看到过很多让我眼前一亮的代码。每当我看到这样的好代码，都会立刻对作者产生无比的好感和认可。且不管这个人处在公司的何种级别，从代码就能看出，他是一个基础扎实的高潜员工，值得培养，前途无量！因此，代码写得好，能让你在团队中脱颖而出。

所以，我的专栏，不仅仅只是讲解设计模式，更加重要的是，我会通过实战例子，手把手教你如何避免刚刚提到的代码问题，告别被人诟病的烂代码，写出令人称道的好代码，成为团队中的代码标杆！而且，写出一份漂亮的代码，你自己也会很有成就感。

3. 提高复杂代码的设计和开发能力

大部分工程师比较熟悉的都是编程语言、工具、框架这些东西，因为每天的工作就是在框架里根据业务需求，填充代码。实际上，我刚工作的时候，也是做这类事情。相对来说，这样的工作并不需要你具备很强的代码设计能力，只要单纯地能理解业务，翻译成代码就可以了。

但是，有一天，我的 leader 让我开发一个跟业务无关的比较通用的功能模块，面对这样稍微复杂的代码设计和开发，我就发现我有点力不从心，不知从何下手了。因为我知道只是完成功能、代码能用，可能并不复杂，但是要想写出易扩展、易用、易维护的代码，并不容易。

如何分层、分模块？应该怎么划分类？每个类应该具有哪些属性、方法？怎么设计类之间的交互？该用继承还是组合？该使用接口还是抽象类？怎样做到解耦、高内聚低耦合？该用单例模式还是静态方法？用工厂模式创建对象还是直接 new 出来？如何避免引入设计模式提高扩展性的同时带来的降低可读性问题？……各种问题，一下子挤到了我面前。

而我当时并没有对设计模式相关的知识（包括设计模式、设计原则、面向对象设计思想等）有太多的了解和积累，所以一时间搞得我手足无措。好在因此我意识到了这方面知识的重要性，所以在之后很多年的开发中，我都一直刻意锻炼、积累这方面的能力。面对复杂代码、功能、系统的设计和开发，我也越来越得心应手，游刃有余。写出高质量代码已经成为了我的习惯，不经意间写出来的代码，都能作为同事学习、临摹的范例，这也成为了我职场中最引以为豪的亮点之一。

4. 让读源码、学框架事半功倍

对于一个有追求的程序员来说，对技术的积累，既要有广度，也要有深度。很多技术人早早就意识到了这一点，所以在学习框架、中间件的时候，都会抽空去研究研究原理，读一读源码，希望能在深度上有所积累，而不只是略知皮毛，会用而已。

从我的经验和同事的反馈来看，有些人看源码的时候，经常会遇到看不懂、看不下去的问题。不知道你有没有遇到过这种情况？实际上，这个问题的原因很简单，那就是你积累的基本功还不够，你的能力还不足以看懂这些代码。为什么我会这么说呢？

优秀的开源项目、框架、中间件，代码量、类的个数都会比较多，类结构、类之间的关系极其复杂，常常调用来调用去。所以，为了保证代码的扩展性、灵活性、可维护性等，代码中会使用到很多设计模式、设计原则或者设计思想。如果你不懂这些设计模式、原则、思想，在看代码的时候，你可能就会琢磨不透作者的设计思路，对于一些很明显的设计思路，你可能要花费很多时间才能参悟。相反，如果你对设计模式、原则、思想非常了解，一眼就能参透作者的设计思路、设计初衷，很快就可以把脑容量释放出来，重点思考其他问题，代码读起来就会变得轻松了。

实际上，除了看不懂、看不下去的问题，还有一个隐藏的问题，你可能自己都发现不了，那就是你自己觉得看懂了，实际上，里面的精髓你并没有 get 到多少！因为优秀的开源项目、框架、中间件，就像一个集各种高精尖技术在一起的战斗机。如果你想剖析它的原理、学习它的技术，而你没有积累深厚的基本功，就算把这台战斗机摆在你面前，你也不能完全参透它的精髓，只是了解个皮毛，看个热闹而已。

因此，学好设计模式相关的知识，不仅能让你更轻松地读懂开源项目，还能更深入地参透里面的技术精髓，做到事半功倍。

5. 为你的职场发展做铺垫

普通的、低级别的开发工程师，只需要把框架、开发工具、编程语言用熟练，再做几个项目练练手，基本上就能应付平时的开发工作了。但是，如果你不想一辈子做一个低级的码农，想成长为技术专家、大牛、技术 leader，希望在职场有更高的成就、更好的发展，那就要重视基本功的训练、基础知识的积累。

你去看大牛写的代码，或者优秀的开源项目，代码写得都非常的优美，质量都很高。如果你只是框架用得很溜，架构聊得头头是道，但写出来的代码很烂，让人一眼就能看出很多不合理的、可以改进的地方，那你永远都成不了别人心目中的“技术大牛”。

再者，在技术这条职场道路上，当成长到一定阶段之后，你势必要承担一些指导培养初级员工、新人，以及 code review 的工作。这个时候，如果你自己都对“什么是好的代码？如何写出好的代码？”不了解，那又该如何指导别人，如何让人家信服呢？

还有，如果你是一个技术 leader，负责一个项目整体的开发工作，你就需要为开发进度、开发效率和项目质量负责。你也不希望团队堆砌垃圾代码，让整个项目无法维护，添加、修改一个功能都要费老大劲，最终拉低整个团队的开发效率吧？

除此之外，代码质量低还会导致线上 bug 频发，排查困难。整个团队都陷在成天修改无意义的低级 bug、在烂代码中添补丁的事情中。而一个设计良好、易维护的系统，可以解放我们的时间，让我们做些更加有意义、更能提高自己和团队能力的事情。

最后，当你成为 leader、或者团队中的资深工程师、技术专家之后，你势必要负责一部分团队的招聘工作。这个时候，如果你要考察候选人的设计能力、代码能力，那设计模式相关的问题便是一个很好的考察点。

不过，我也了解到，很多面试官实际上对设计模式也并不是很了解，只能拿一些简单的单例模式、工厂模式来考察候选人，而且所出的题目往往都脱离实践，比如，如何设计一个餐厅系统、停车场系统、售票系统等。这些题目都是网上万年不变的老题目，几乎考察不出候选人的能力。在我的专栏中，有 200 多个真实项目开发中的设计模式相关问题，你跟着看下来，足以让你成为设计模式方面的大牛，再来面试候选人的时候，就不用因为题目老套、脱离实践而尴尬了！

重点回顾

今天，我们讲了为什么要学习设计模式相关的知识，总结一下的话，主要有这样五点：应对面试中的设计模式相关问题；告别写被人吐槽的烂代码；提高复杂代码的设计和开发能力；让读源码、学框架事半功倍；为你的职场发展做铺垫。

投资要趁早，这样我们才能尽早享受复利。同样，有些能力，要早点锻炼；有些东西，要早点知道；有些书，要早点读。这样在你后面的生活、工作、学习中，才能一直都发挥作用。不要等到好多年后，看到了，才恍然大悟，后悔没有早点去学、去看。

设计模式作为一门与编码、开发有着直接关系的基础知识，是你现在就要开始学习的。早点去学习，以后的项目就都可以拿来锻炼，每写一行代码都是对内功的利用和加深，是可以受益一整个职业生涯的事情。

课堂讨论

今天课堂讨论的话题有两个：

1. 聊一聊你对设计模式相关知识的重要性的看法；
2. 在你过往的项目开发中，有没有用过某种设计模式？是在什么场景下应用的？解决了什么问题？

欢迎在留言区发表你的观点，积极参与讨论。你也可以把这篇文章分享给你的朋友，邀请他一起学习。

AI智能总结

学习设计模式对程序员来说至关重要。首先，掌握设计模式知识可以提高面试成功率，成为团队中的代码标杆。其次，设计模式能够提高复杂代码的设计和开发能力，帮助程序员写出高质量的代码。此外，学习设计模式还能为职业发展做铺垫，让读源码、学框架事半功倍，提高职场竞争力。因此，学习设计模式不仅能提高开发能力，还能在面试和职业发展中获得更多机会。

全部留言 (671)

最新　精选

仰望星空

2019-11-04

老师能不能讲讲函数式编程思想，设计模式都是基于面向对象的，而现在更流行函数式编程。

作者回复: 函数式编程感觉还是没面向对象流行 所以专栏中没讲

共 3 条评论>

20

时光之刃

2019-11-05

希望老师能举一些开源代码中的设计模式，比如netty或者ES等

作者回复: 可以的 我后面可以考虑多写几篇加餐

11

mgs2002

2020-11-29

30多了还来得及吗。。

作者回复: 来得及啊 加油

共 2 条评论>

4

蓝二哥哥我才是无羡啊...

2019-11-11

做服务端开发相关工作，对于一个初入职场的小白来说，什么工厂模式单例模式，只有简单的知道概念，看项目都不能明白他们的体现，但最近发现了单例模式在装饰器方面的应用是真的让人印象深刻，从中体会到了编程的乐趣，所以下定决心要好好学习设计模式！

还有老师说的：“你以为你读懂了别人的代码，其实你可能还未领悟到其中的精髓！”我思考一阵，发现好像真的是这样，最近在学习前辈的代码阶段，好像只是懂了代码实现的业务逻辑，并不十分明白为什么要这样去实现，好处在哪，所以想问问老师，怎么才能领悟到别人代码的精髓呢

作者回复: 内功足够扎实，才能领会别人的代码为什么这么写。

4

Geek_5a5d9a

2020-11-13

移动端开发:
常用的设计模式: 单例模式, 代理模式, 观察者模式, 适配器模式, 还有用到一些策略模式

作者回复: 嗯嗯

1

闪耀之作r
2019-11-06

设计模式什么时候学最好？需要什么基础，有先后顺序分吗？数据结构与算法怎么和设计模式相结合，设计模式不只是java才有，其他的也有，其他的都需要单独学习吗

作者回复: 现在学最好，会编程即可。两者不需要结合。不需要单独学，设计模式跟编程语言没太大关系。

1

李奇峰
2020-11-29

设计模式在大学的时候通过《Head First 设计模式》这本书学过一段时间，但是工作之后就忘掉了。好像都只是能把业务逻辑完成就可以了，很少用到设计模式的思想去编写代码。

作者回复: 加油~~~

高崇波
2020-11-28

我是做嵌入式开发的主要是用C语言，C语言语法结构上不支持面向对象，但是在思想可以实现面向对象，对这边了解的太少，代码的架构能力太薄弱，每次做项目都是考虑实现功能，没有对代码的扩展性，可维护性等做太多的考虑。希望学习设计模式能够带我提升一个档次

作者回复: 嗯嗯 加油

峰
2020-11-21

设计模式是组织代码的艺术，怎么让接盘侠研究代码的时候能专注于一个更小的上下文，怎么面对不断变化的需求，隔离变化，能做可扩展的修改，或者尽量小范围修改。
还记得，那是一个查询路由的东东，对不同查询类型，都会走一溜烟的处理链条，而在链条的每一环，都要判断查询的类型是什么要不要跳过，或者进行特殊化的处理。这样对新增一个查询类型，就意味着要改整个链条。我做的修改是，把查询类型和查询链条做了接耦，为每一种查询装配一种独特的处理链。

作者回复: ������


斐波那契
2019-11-06

设计模式的重要性不用多说 自己体会就好 但是烂代码真的很神烦 老师 现在刚接触一个项目 我个人感觉写的蛮烂的 开发起来各种坑 迭代时间紧 请问 怎么破

作者回复: 加餐文章有讲到 别急