

## 开篇词 | 你写的每一行代码，都是你的名片

2019-01-02 范学雷



讲述：黄洲君

时长 07:13 大小 9.92M



你好，我是范学雷，现在是 Oracle 的主任工程师，也是 OpenJDK 和 Java 安全的评审成员。很高兴和你一起聊聊怎么写好代码这个话题。

我第一次接触计算机，是在 1994 年。那时候，我还是大学一年级的一枚青瓜。当时的计算机发展，正处于青涩的少年阶段。“互联网”也还是一个非常生僻的名词。当时，我们用的计算机是“286”，操作系统是 DOS，编程语言还是 Fortran 和 C 语言，Java 语言还没有正式诞生。每次上课，都要随身携带容量为 360KB 的 5.25 英寸软盘。娇气的软盘啊，可是不好伺候，动不动就损坏。那时候最渴望的事情，就是能有一张存储容量高达 1.44MB 的高密度 3.5 英寸软盘。

计算机启蒙课给我印象最深的是什么呢？不是怎么写程序，而是不停地折腾软盘，一直重复“修复、备份”这个过程。也许是因为软盘的拷贝和修复太无聊，我一直对计算机以及编程没有特别大的兴趣。

但大学最后一年，两件小事让我改变了对计算机和编程的态度，给我带来了巨大的影响。

第一件事是，我一个同学编写的五子棋人机对弈程序，当时打遍全班无敌手。厉害吧！用现在的话说，就是“怎么可以这么炸”！这可不是使蛮力，用穷举法就可以搞定的，到底是怎么做到的？这引起了我对计算机程序的强烈兴趣。

第二件事是，我另一个同学的毕业论文选择了密码学作为研究方向。这个同学有一个优点，不管什么事情，都特别喜欢分享。用东北话说，就是爱嘚瑟，逢人便絮叨。最后差不多全班都知道了密码学的一些基本概念，了解了与之相关的好多传奇故事。密码学这种超神秘、超有趣、超复杂的存在，简直吊足了我的胃口。

**我们的每一次经历，都塑造着我们自己。**写人机对弈程序的同学，第一次面试就找好了工作，进了最好的公司。研究密码学的同学，是中国商业密码产业化最早的参与者之一。而我自己呢，在他们的影响下，也找到了计算机的乐趣，享受着解决复杂问题带来的喜悦。

编程和密码学这两个东西合在一起，就是我现在每天工作的主要内容。具体来说就是两件事：**写代码和看代码。**

写代码这件事，就我自己的经历来说，有点像过山车。

我刚开始学习编程时，写几十行代码都觉得痛苦、费劲，不知道从哪儿下手。这种状况一直持续了很多年，直到 1998 年我参加工作，编写程序成了我的职业。职业也就意味着，编码有了具体的目标，代码有了具体的质量要求。

我是幸运的。目标，有人掰碎了、揉烂了给我讲；质量，有人睁大了眼睛盯着看，也有人不顾情面地给我指出各种问题。有了目标就有了思路，有了要求就有了动力。如果再有人不离不弃地帮助，每一个度日如年的煎熬，最终都会变成“士别三日”的惊喜。慢慢地，我就可以写几百行、几千行、几万行甚至十几万行的代码了。而且越写越快，越写越好。

大概到了 2000 年的时候，代码设计对我来说可能依然很费时间，但是只要写起代码来，一天数千行也是很常见的。一天洋洋洒洒写数千行代码，暗暗觉得自己挺牛，挺了不起的。

**“无知要比知识更容易产生自信”**。幸运的是，这种盲目的自信没有持续太久，我很快就见识到了更宽阔的世界。2004 年，我加入了 **Java 安全组**，真正地见识到了，优秀的设计

和优秀的代码，是怎么一步一个脚印地出炉的，了解到代码背后的各种综合考量和艰难取舍。慢慢地，我自己也完成了从“代码数量优先”到“代码质量优先”的思路转变。

如果回头看十多年前编写的代码，就像是看筛子一样，到处都是清清楚楚的破洞。也许，这是每个程序员都要经历的过程吧。

**我们总是先要解决掉数量问题，然后才能解决掉质量问题。**

这个过程，还真的有点“看山是山，看山不是山，看山还是山”的味道。

看代码这件事，对我来说，其实是一个收获大于付出的过程。

OpenJDK 的代码必须通过评审才可以提交。OpenJDK 社区有非常广泛的代码贡献群体，有些是还没有毕业的年轻学生，也有些资深的业界老专家。新手当然有新手的困惑，而老辣的程序员，也会犯简单的错误。

一个代码评审者的主要工作，不是批准或者拒绝提交的代码，而是提出合理的建议，帮助代码提交者规避这些失误或者错误，编写出更优秀的代码。

看代码看得多了，对代码就有更多的了解。比如，**什么样的代码更容易出问题？什么样的代码会招惹麻烦？什么样的代码出力不讨好？什么样的代码小问题闯大祸？**

同时，也对程序员有了更多的了解。比如，**为什么我们不愿意写注释呢？为什么代码写完就不愿意修改了呢？为什么我们不愿意做测试呢？为什么我们向往自由而不愿意遵守规范呢？**

每一行代码，都体现着程序员的修为，思考问题的深度，甚至是处理问题的习惯和态度。**代码，是我们交流的语言和处世的名片。**

这些问题，思考总结下来，就是代码评审的经历馈赠给我的礼物，而且是天大的礼物。

现在我把这份礼物沉淀下来，就是我们这个专栏的主要内容。我想通过这样一个专栏，让你拥有和我一样的收获。

# 《代码精进之路》课程目录

## 认知篇

- ① 从条件运算符说起，反思什么是好代码
- ② 把错误关在笼子里的五道关卡
- ③ 优秀程序员的六个关键特质

## “规范”修炼篇

- ④ 代码规范的价值：复盘苹果公司的GoToFail漏洞
- ⑤ 怎么给你的代码起好名字？
- ⑥ 代码整理的关键逻辑和最佳案例
- ⑦ 写好注释，真的是小菜一碟吗？
- ⑧ 写好声明的“八项纪律”
- ⑨ 怎么用好Java注解？
- ⑩ 异常处理都有哪些陷阱？
- ⑪ 组织好代码段，让人对它“一见钟情”
- ⑫ 组织好代码文件，要有“用户思维”
- ⑬ 接口规范，是协作的合约
- ⑭ 怎么交付代码的用户指南？
- ⑮ 编写规范代码的最佳实践清单
- ⑯ “规范”模块热点问题答疑



## “经济”修炼篇

- 17 导读：为什么说你要关注代码的性能问题？
- 18 思考框架：什么样的代码才是高效的代码？
- 19 少即是多，该怎么避免过度设计？
- 20 简单和直观，是永恒的解决方案
- 21 怎么设计一个简单又直观的接口？
- 22 高效率，从超越线程同步开始！
- 23 怎么减少内存使用，减轻内存管理负担？
- 24 黑白灰，理解延迟分配的两面性
- 25 使用有序的代码，调动异步的事件
- 26 怎么让代码“少招惹麻烦”？
- 27 怎么让代码“可持续发展”？
- 28 怎么尽量“不写”代码？
- 29 编写经济代码的最佳实践清单
- 30 “经济”模块热点问题答疑

## “安全”修炼篇

- 31 为什么安全的代码这么重要？
- 32 如何衡量安全的代码？
- 33 整数的运算有哪些安全威胁？
- 34 怎么处理数组和集合的安全陷阱？

- ③5 怎么处理敏感信息？
- ③6 继承有什么安全缺陷？
- ③7 怎么处理跨界数据的安全隐患？
- ③8 对象序列化的危害有多大？
- ③9 怎么控制好代码的权力？
- ④0 怎么用好Java的内在安全机制？
- ④1 怎么保持代码的长治久安？
- ④2 怎么防范软件安全风险？
- ④3 编写安全代码的最佳实践清单
- ④4 “安全”模块热点问题答疑
- ④5 尾声：如何成为一个编程好手？

回顾我这二十多年的程序员经历，我觉得自己是非常幸运的。现在，我们常常调侃“35 岁码农大龄恐惧症”。幸运的是，当这种病毒一般的焦虑开始流传开来的时候，我早已经过了 35 岁，已经来不及担心了。

这种焦虑之所以广泛流传，背后传达的一个本质问题就是：**作为一名软件工程师，我们该怎么快速成长，并且保持长久的竞争力？**

这个问题的终极方法，只有一个，那就是**持续地交付优秀的结果**。宜早不宜迟。

作为解决现实问题的软件工程师，不管资历深、资历浅，我们都需要编写优秀的代码，并且是越来越优秀的代码，因为这是我们生存的基本依靠。**作为活在现实世界的技术工程师，我们需要保持长久的竞争力，甚至是越来越强的竞争力，因为这是改善我们生存质量的最好方式。**

在这个专栏里，我会带着你开始一段代码精进的旅程。和你一起来看一看、摸一摸那些年别人踩过的坑，来聊一聊、试一试我们的代码可以写得有多棒，享受这个打怪升级的过程。

那么现在，**给你一个机会，你敢不敢吐槽一下你见过的或者写过的，最“差劲儿”的代码？或者，你愿不愿意秀秀你自己最中意的代码？**

也欢迎你在留言区写下自己的编程故事，等到专栏结束后，我们再回过头来，看看你走出了怎样的成长轨迹。

我渴望做那些伟大而高贵的任务，但是，我首要的责任和快乐却是去完成那些卑微的任务，把它们也当作伟大而且高贵的一样。世界在前行，不只是那些英雄们的力量在推动，也同样包括那些来自每个诚实的工作者微小推动的积累。—— 海伦·凯勒



# 代码精进之路

你写的每一行代码都是你的名片

范学雷

Oracle 首席软件工程师  
Java SE 安全组成员  
OpenJDK 评审成员



新版升级：点击「👤请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得转载

下一篇 01 | 从条件运算符说起，反思什么是好代码



李英权

2019-01-04

12

JAVA编程15年，最近终于理解什么是oo所谓的高内聚低耦合：高内聚就是让同一个level的逻辑出现在同一个class 而不是a调b b调c的伪分层 乱调用；低耦合就是用分层的抽象来隔离变化的细节与不变的算法。这也是所谓的开闭原则——向扩展实现细节开放 向破坏算法流程关闭。

归纳起来，其实就是template method设计模式。设计模式有那么多种，我用下来就这...  
展开

作者回复: 好经验。后面我们会聊到接口设计，拜托到时候多分享分享你的经验。



Change

2019-01-02

11

已剁手，通过学习，期待自己的编码技术更精进！

展开

作者回复: 我们一起加油



Being

2019-01-02

8

最初的在编码方面的成就感就是工作之初每天回家学设计模式，然后在一次开发中，用状态模式减少了一大堆switch case，便于理解，也便于后期维护，自此对写出优秀代码的思考一发不可收拾。有时也会面临设计过度的窘境，这也是个要“修炼”的过程呀。我现在就是那种只要有思路，一天可以一两千行的输出，但也渐渐开始追求精炼，今天还在琢磨怎么设计好合适的接口。...

展开

作者回复: 啃设计模式，也是我的一段经历。设计模式可以帮助我们快点树立接口思路。



VAE

2019-01-03

4



一直以来代码都写的很烂，代码中充斥着大量的if-else，我也知道很烂，但是又不知道如何优化，希望能从这个专栏里面有所收获。

作者回复: 😊我猜，一定是没有人逼着你一个方法不能超过45行。



sophia

2019-01-03

👍 3

我是一个强迫症，有代码洁癖。期待中。期盼能写出更多好可读和高性能的代码。😊

展开 ▾

作者回复: 好习惯，也期待分享你的经验。



五花肉小生

2019-01-02

👍 3

逻辑不清、命名意义不明、拼写错误、大量重复代码、耦合度奇高等等，有幸在我工作的第一年就看见了，有些人靠忽悠，这种情况真的存在。

作者回复: 很多人没有学过这方面的知识，希望编程语言老师多讲讲编码规范，这样大家都会更好。



大帅哥

2019-01-03

👍 2

工作六年，每次看见之前写的代码都想去优化，优化完过段时间再来看还想优化的冲动，希望通过此专栏的学习提高编码质量和优化思路

作者回复: 就是这样，优化无止境😊，旧代码，let it go



李小辰✧

2019-01-03

👍 2

## 最差的代码就是我自己写的代码

展开 ▾

作者回复: 专栏有很多练手题, 😊让我们看看会有多不好啊。不过, 我觉得对自己代码不满意的, 一般不会差。



pyhhou

2019-01-03

👍 2

已购买, 作为一个刚从学校出来, 非计算机专业, 没多少码代码的经验的年轻人, 在工作当中总是会因为一些写代码的细节问题而出现各种各样的问题, 例如如何写好一个web API, 以及documentation, 希望自己能在这个专栏中学到的东西, 也请老师指点迷津, 以下是列的一些期望, 不对的地方还请老师指出:

1. 重新认识写好代码这件事...

展开 ▾

作者回复: 我也不是计算机专业的, 所以要比别人晚一点, 不过, 也就晚一点点。你关心的几个问题, 就是我们这个专栏的目标。差不多十篇文章的时候, 你看看我们有没有解决好这些问题, 到时候给我个反馈啊。

知识和认识, 要转化成能力, 需要大家一起讨论、争论。欢迎你跟进练手题, 参与讨论。这个专栏, 大家参与度越高, 收获就会越大。



科哥

2019-01-02

👍 2

工作两年, 看了很多编程方面的知识, 纠结过、迷惘过也兴奋过。但是还是觉得自己的代码量不够, 写出来的代码也可圈可点, 希望能够通过自己多练以及这个专栏的帮助能够写出漂亮的代码。

展开 ▾

作者回复: 专栏里有不少练手的题目, 我们一起来练练手, 把专栏文字变成自己的能力。



liu

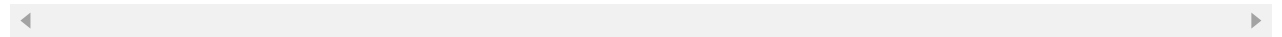
2019-01-02

👍 2

已购买。期待精彩

展开 ▾

作者回复: 欢迎参与以后的讨论



**A1李自飞**

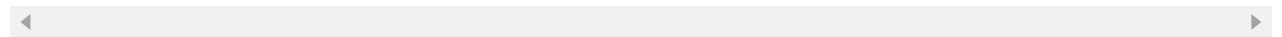
2019-01-02

👍 2

老程序员一枚

展开 ▾

作者回复: 哈哈😊，一起来切磋交流



**阿官(^·...**

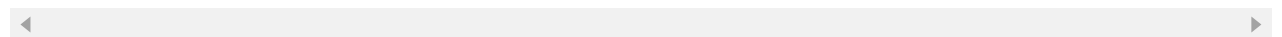
2019-01-03

👍 1

以前实现业务逻辑，代码里确实很多判断，最恶心的是嵌套三层if else的代码，可读性真的很差，不跑个单元测试根本很摸清具体的逻辑。感谢作者，希望通过本专栏对做业务开发的我有所帮助

展开 ▾

作者回复: 测试很重要，但是测不完所有的逻辑，代码本身逻辑清楚才好。



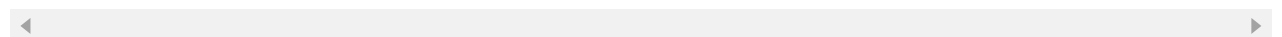
**大鹏展翅**

2019-01-03

👍 1

已买，公司正好正在搞编码规范，生死存亡的关键时刻，大家都在学习如何写出优秀的代码，希望有所帮助！

作者回复: 赞！写好代码，省钱省时间，大家成长也快。跑得慢，到得快。



**Alexdown**

2019-01-03

👍 1

已入手，看过《代码大全》（尽管我觉得该叫代码完成或完成代码），《代码整洁之道》等。成功应用过一些书中的原则，比如用描述性函数名代替注释，也有一些争议的原则让我挣扎，函数的入口参数是否每个都要校对数据类型（python），取值范围并分别给予提示，提示方式用异常还是空等等。在什么场景下，选择什么方式，很没谱也没人能讨论

...

展开 ∨

作者回复: 我喜欢极客时间的一点就是我们有讨论区。有问题就塞讨论区里☺。看文字进步慢，讨论、练手，一起来学的才快。

◀ ▶



昵称而已，...

2019-01-03

👍 1

期待中，四年了，自己写的代码总觉得不是很好！

展开 ∨

作者回复: 我想，应该可以帮到你。这个专栏有点像案例课，多参与多分享。

◀ ▶



绝影

2019-01-03

👍 1

我们总是先要解决掉数量问题，然后才能解决掉质量问题。

展开 ∨

作者回复: 哈哈，像走路，先歪歪的站住，再走，再跑，再攀登。

◀ ▶



vector

2019-01-02

👍 1

每次写代码总会刻意设计，每次又都不满意，总感觉有更好的实现方式。期待老师的分享，代码写诗，码无止境。

作者回复: 赞，无止境👍

◀ ▶





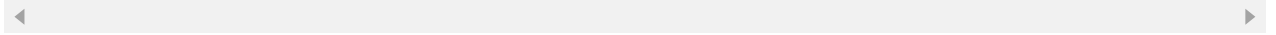
Geek\_5b3cc...

2019-01-02

👍 1

希望会涉及业务功能拆分。一个业务模块单纯的三层架构会让service变重，并且一个业务会涉及其他业务的。一些根据返回值，查询并为新对象赋值操作，提取出来不知道该放到哪里。

作者回复: 嗯，老大难的问题。后面会讨论有一个我常用的解析问题办法，不知道到时候是不是对上。一起期待吧。



正在减肥的...

2019-01-02

👍 1

工作了6年，每次写代码都觉得自己写的代码很烂。最近这年这种想法越来越强烈。先在有时间就去看各种框架的源码。希望老师能把你多年的宝贵经验传授给我们。

作者回复: ☺我也需要你传授经验给大家。多使用讨论区，练手题，我们一起学，一起变的更好。

