

# Java 进阶

作者：草原上唱山歌

链接：<https://www.zhihu.com/question/654003758/answer/3508020193>

来源：知乎

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

**说一下我看过的一些比较不错的代码，看了对Java能力一定会有质的提升，不敢说已经学明白了**

国人写的**模板引擎**，代码质量很高：

subchen/jetbrick-template-2x

**数据结构**和**算法**类的，同时有测试代码：buptdavid/datastructure|kdn251/[interviews](#)

Java写的高性能数据库：jankotek/mapdb

**Netty**实现的高性能RPC，有写博文分析：tang-jie/[NettyRPC](#)

Java编写的原子组件库，内功非常深厚：mindwind/[craft-atom](#)

前点评网的一位大牛**黄艺华**写的爬虫框架：code4craft/webmagic

针对Java8的一系列代码示例，纯干货无废话：winterbe/java8-tutorial

Java**设计模式**的总结以及代码和文章：

iluwatar/java-design-patterns

最优雅的网络请求库，一个Java文件：kevinsawicki/http-request

Java程序员一个 交流社区：[https://jq.qq.com/?\\_wv=1027&k=5pkxbG4](https://jq.qq.com/?_wv=1027&k=5pkxbG4)

以上是我看过质量都还不错的代码库，难度相对不是特别高，但对于编程思想的提升还是有很大帮助，互联网资源如此丰富的时候我们更需要的是静下心来安静学习和思考而不是一味的追求信息大爆炸，所以我只挑选了对于代码能力有提升的难度不是特别高的分享给各位。

同时也推荐阮一峰的《[如何变得有思想](#)》一书，这本书记得在3年前看的时候对于个人的编程和人生起了一定的改善作用，如果你觉得对你有帮助也可以尝试看看，以上是我的一些个人见解。

最早看的 [架构探险](#) 从零开始写Java Web框架，[黄勇](#)写的，算是一种启蒙，作者自己写了一套ioc和aop框架以及[mvc请求分发框架](#)。

跟着写了一遍，基本明白了Spring 和[SpringMVC](#)是怎么工作的

然后顺着思路看了2遍Spring和SpringMVC的源码，一遍是跟着流程走了一边，一遍是基于接口和类的功能整体了解了一遍架构设计

再后来看了[how tomcat works](#)，看了一遍中文的，时隔一年又看了一遍英文的，tomcat的作者亲自讲解tomcat的进化，怎么从一个几十行的弱鸡http服务器变成一个servlet容器的，跟着书和各种源码把tomcat的演变写了一遍，真正的源码也过了一遍，但是分支太多了

作为一个Java web 开发者，我觉得从tomcat的启动到tomcat接受http请求到SpringMVC路由分发请求，到业务代码处理，到模板渲染，到返回对应的http信息都应该了解对应的过程，这样遇到很多问题都能第一时间判断出对应的位置，让一些'很神奇'的问题不再神奇

然后就是netty了，看了一遍异步的事件驱动reactor模型的nio服务器实现，对异步，多线程，nio理解上了一个新的台阶

中间穿插读了util包的各种类

Spring几乎已经不需要解释了，用到的都应该去阅读一下，无论是设计思想，代码规范，还是设计模式，接口设计，类加载，都是非常优秀的源码

tomcat太大了，并不是很推荐阅读，源码对于阅读者也不是很友好，但是how tomcat works还是很值得阅读的

netty本身同样很优秀，但是更重要的是实现的思想

总之阅读源码有什么感觉呢，就好像打开了一扇通往更大世界的大门，能看到厉害的代码技巧，健全的错误异常机制，更重要的是对已有问题的解决方案，和解决问题的思路，这些都是一个程序员成长过程中最好的教材。

阅读源码，有时候也需要有一定的技术基础和耐心。假如你才刚刚入门Java，就立即去啃《深入Java虚拟机》这类书，别人可能觉得很好，未必适合初学的你。

如果不带着思考的问题，读源码的作用可能没有想象中那么大。

以下可能是阅读源码的目的：

- 1、遇到框架使用的问题，解决起来很轻松。
- 2、学习别人的架构设计，在自己设计产品时，可以模仿应用。
- 3、源码阅读也是一条提高自身技术能力的很好的途径。
- 4、培养代码规范，如格式、命名。开源框架的源码一般都很优雅。

例如Spring作为优秀的框架之一，被程序员广泛的应用。

我在面试Java研发人员的过程中发现，有些同学说自己看过Spring的源码，就让他随便聊聊Spring，就蒙了，汗。

阅读Spring源码后，对于Spring有哪些核心的组件？Bean如何解析？资源类如何访问？AOP是如何运用这些来完成工作的？Spring都是用了哪些设计模式来完成了这些功能等？应该有一些自己的思路 and 想法。

Spring通过抽象复杂多变的对象，进行规范，根据它自己的规范设计一个容器，用容器来构建维护它们复杂关系。

可以去阅读Spring源码的同时，了解它的架构，分析它的设计理念，找出好的设计思路，可以为以后我们的程序设计带来很多思路。

这样你对Spring就有个清楚的认识，构建优雅代码上也有一些心得，在工作中很多场景都可以有相应的应用。

jdk就是最好的java源代码，作者为了性能也是拼了，里面会考虑对gc友不友好，生成的内存屏障指令要尽量轻量级，为了应对伪共享做填充，能用位移的就用位移，不说了，我继续看源码去了。

最近一直有在看JDK源码注释，主要是深入研究下基础知识和并发编程。结合注释看源码，得到的细节非常多。经常可以看到一行代码上有三十四行注释，会把一个方法的使用示例，需要注意的问题，方法实现细节说明等等，全都说的很清楚。比如Object的wait和notify方法。注释里说wait方法会让当前线程，被放入到一个set中，这个set是和当前对象的监视器关联的。所以这就说明了为什么wait方法为啥需要在synchronized方法里调用，因为要获得对象的监视器嘛。还有就是线程进入等待后，会让渡出当前对象锁，这个是和线程睡眠的区别。还说明了线程假醒的问题，因此推荐把wait方法放在while循环里。看完后感觉真是干货满满，扫除了很多基础方面的知识盲点，受益良多。