

加餐六 | 什么才是所谓的编程能力？如何考察一个人的编程能力？

王争 · 设计模式之美



在招聘要求里，我们经常看到“要求候选人有扎实的编程能力”。在面试反馈中，我们也经常看到面试官用“编程能力很强”来评价候选人。那到底什么是编程能力呢？如何考察一个人的编程能力呢？又如何提高编程能力呢？今天，我们就编程能力这个话题展开聊一聊。

话不多说，让我们正式开始今天的内容吧！

什么是编程能力？

所谓编程能力，指的是把“逻辑”（或者说“功能”“业务”）翻译成代码的能力。所谓编程能力强，指的是，不仅能编写正确的代码，而且编写代码的速度很快，写出来的代码 bug 很少、性能很好、质量很高。

更加具体点讲，一个编程能力强的人，能熟练使用编程语言、开发类库等工具，思路清晰，面对复杂的逻辑，能够编写出 bug free 的代码，能够合理地利用数据结构和算法编写高效的代

码，能够灵活地使用设计思想、原则和模式，编写易读、易扩展、易维护、可复用的高质量代码。

相反，一个编程能力差的人，往往逻辑思维能力比较差，面对复杂的逻辑，编写代码的速度很慢，而且容易考虑不周，写出来的代码 bug 很多，更没有性能意识，不懂得如何分析代码的时间复杂度、空间复杂度，更不懂得如何借助现成的数据结构和算法来优化代码性能。除此之外，写代码的时候，几乎不考虑代码的可读性、可扩展性等质量问题，只追求能运行就可以。

如何考察编程能力？

前面我给出了编程能力的简单的定义，从定义中，我们能很清楚地了解，考察一个人的编程能力的几个要素。总结一下，我觉得主要包含这样三个方面：编程语言，数据结构和算法，设计思想、原则和模式。

考察编程能力，那就离不开写代码。所以，在面试中，我一般都会出道编程题，让候选人写段代码看看，也就是所谓的“白板编程”。白板编程在外企面试中比较流行，国内有些候选人不怎么能接受，特别是工作年限比较长的候选人，一听说要写个代码，就觉得这是在“羞辱”他，觉得不应该从这么基础的东西考起。

不过从我多年的面试经验来看，这种拒绝写代码的“大龄码农”，满嘴“架构、高可用、高并发、分布式”，往往代码写得惨不忍睹。所以，只要是应聘一线技术研发岗的候选人，不管是资深工程师、架构师，还是技术 Leader，我都会要求他现场写一段代码。因为这是最直接、最有效检验这个人基本技术素养的途径。

一般来讲，编程语言都可以快速掌握，所以，我一般都不会把它拎出来单独考察，只是顺带着考察一下就可以了。我会重点考察后两个方面：数据结构和算法，设计思想、原则和模式。但是，要想设计一个题目，既能考察到这两方面的知识，又能在不到 1 个小时的面试时间内完成，还是挺难的。所以，对于这两个方面，我一般都分开来考察。我今天重点讲对数据结构和算法的考察，对于设计思想原则和模式的考察，我后面有文章专门来讲。

对于数据结构和算法的考察，我个人不建议面试题目涉及需要记忆的算法，比如被很多人诟病的面试题：写个快排。没有人会天天背诵快排算法，候选人写不出来也理所应当。如果我们换个问法，比如给候选人讲一下快排的思想，然后让候选人用代码实现，测试候选人的代码翻译

能力，我觉得这反倒是一个比较好的面试题。除此之外，我也不建议面试题目涉及特殊解题方法或技巧，比如需要用到线段树、并查集等高级数据结构。毕竟大家在工作中不常用到这些知识，不知道或者忘记了我觉得也很正常。

所以，那种不依赖很强的算法背景和特殊的解题技巧的题目，比较适合用来面试。比如像下面这样的面试题就比较好：“写一个函数将 IPv4 地址字符串（仅包含数字、点、空格）转化成 32 位整数。另外，数字和点之间的空格是合法的，其他情况均为非法地址，要求输出合法地址的 32 位整型结果。”

我觉得这种题目是比较公平的，对于没有刷过题的人来说也很友好，因为它纯粹是在考察候选人的基本编程素质：逻辑思维是否清晰，细节是否考虑全面，是否能写出 bug free 的代码，是否有计算机思维，会关注时间空间复杂度等。

如何提高编程能力？

刚刚我们讲了什么是编程能力，如何考察编程能力，现在，我们讲下如何提高编程能力。实际上，我的两个专栏《数据结构与算法之美》《设计模式之美》，就是为了提高你的编程能力而设计的。《数据结构与算法之美》专栏教你如何写出高性能代码，《设计模式之美》专栏教你如何编写易读、易扩展、易维护的高质量代码。

方向很明确，但是要真的提高编程能力，光学不练肯定是不行的。

对于数据结构和算法的练习，我们推荐你多刷 LeetCode 上的题目。刷题不仅仅能加强你对数据结构和算法的掌握，还能锻炼你的逻辑思维能力、写出 bug free 代码的能力、快速实现复杂逻辑的能力，也能锻炼你的性能意识。所以，刷题不只是为了面试，刷题对这些能力的锻炼，都有助于你在平时的业务开发中写出好的代码。

对于设计思想原则和模式的练习，它就不像算法练习那样有现成的题库了。所以，要在平时的开发中进行练习。比如，拿到一个功能需求的时候，我们先去思考一下如何设计，而不是上来就写代码。写代码时，我们时刻思考代码是否遵循了经典的设计思想、设计原则，比如是否足够可扩展、是否满足 SOLID 原则、可读性如何等等。写完代码之后，我们再思考一下，代码是否有进一步优化的空间。做 Code Review 的时候，看到别人的优秀的代码，我们就去思考一下，有哪些值得借鉴的地方。总之，在平时的开发中，我们要刻意地去做这种跟代码质量、

代码设计相关的思考训练。时间长了，这种思考就能成为习惯、本能反应，慢慢地，你的代码能力也就不自觉地提高了。这一部分内容你可以回过头去再看下第 100 篇，我们前面有非常详细的讲解。

课堂讨论

实际上，不管从事什么行业，要积累的东西都可以分为“变”与“不变”两类。“不变”的是内功，“变”的是招式。我们要善于发现、持续积累那种“不变”的能力，而不是要去盲目追逐一直都在“变”的招式。除了编程能力之外，在 IT 技术领域，你觉得还有哪些不变的内功？

欢迎留言和我分享你的想法。如果有收获，也欢迎你把这篇文章分享给你的朋友。

AI智能总结

编程能力是指将逻辑转化为代码的能力，包括编写正确、高效、高质量的代码。考察编程能力主要包括编程语言、数据结构和算法、设计思想、原则和模式。面试中通过白板编程考察候选人的基本技术素养，重点考察数据结构和算法。建议不考察需要记忆特定算法或技巧的题目，而是考察基本编程素质。提高编程能力可通过刷 LeetCode 题目加强对数据结构和算法的掌握，锻炼逻辑思维、写出 bug free 代码的能力，同时在平时开发中练习设计思想、原则和模式。文章强调内功的重要性，提倡持续积累不变的能力。

© 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 (37)

最新 精选



Jie

2020-07-01

沟通能力很重要，和领导和同事和客户可以聊得起来。和机器打交道久了，人不能变得像机器一样。



57



J.Smile

2020-07-01

做人要务虚，做事要务实。程序员不懂得务虚在职场很吃亏，即便技术不差，但得出类拔萃，挺难。



君哥聊技术

2020-07-06

技术上讲，数据结构，操作系统，网络，设计模式，数据库，这些都是不变的内功



👍 16



enjoylearning

2020-07-01

产品意识也是一种内功，不能只是还原客户的需求，要了解客户需求背后的需求。



👍 16



守拙

2020-07-01

“写一个函数将 IPv4 地址字符串（仅包含数字、点、空格）转化成 32 位整数。另外，数字和点之间的空格是合法的，其他情况均为非法地址，要求输出合法地址的 32 位整型结果。”

思路：

1. 根据.和空格判定输入是否合法；
2. 将输入以.分割为4个8bit的十进制int值；
3. 遍历4个int值，将其转为二进制值；
4. 拼接4个二进制值，得到32位整型结果。

共 4 条评论 >

👍 14



程序员小跃

2020-10-24

最近一直在和同学聊如何提升自己的能力，因为他刚转行到 Java 开发不久，我零零碎碎的整理了一些知识点给他；恨自己没早点学习专栏，没有把这篇文章做一个观后感的整理。

还早现在还来得及，我反复读了几遍，顺带分享给我同学，希望他能转行找到心仪的工作。



👍 7



业余爱好者

2020-07-01

量子计算普及之前，冯诺依曼架构的理论知识（如os,网络等）还是需要学习的，不过这些底层知识的学习最终还是要落到编程能力上，因为少有人去设计一个操作系统或编译器。

编程能力确实是一个衡量程序员水平的很好的指标，编程能力又可以拆分为算法能力，设计能力等。对程序员来说，一切工作都是围绕编程的，失去这一宗旨，一切都是扯淡，即使所谓沟通之类能力也是为了协作以完成编程目标而已。



7



Geek_7f18eb

2020-07-03

```
private static String ipTo32(String str) {
    String result = "";
    int length = str.length();
    int i = 0;
    for (; ; ) {
        if (i > length - 1) {
            if (result.length() < 32) {
                result += "0";
                continue;
            } else {
                break;
            }
        }
        char index = str.charAt(i);
        Boolean last = Character.isDigit(index);
        Boolean lastPoint = ".".equals(String.valueOf(index));
        if (last) {
            result += String.valueOf(index);
        }
        int j = i + 1;
        if (j < length && ".".equals(String.valueOf(str.charAt(j)))) {
            Boolean nextPoint = ".".equals(String.valueOf(str.charAt(j + 1)));
            Boolean next = Character.isDigit(str.charAt(j + 1));
            if ((last && nextPoint) || (lastPoint && next)) {
            } else {
                result = "false";
                break;
            }
        }
        i++;
    }
}
```

```
}  
    return result;  
}
```



6



我能走多远

2020-07-01

底层得知识是内功，计算机理论基础；内核的内存管理及文件管理；数据结构与算法等等吧

共 3 条评论 >



4



全炸攻城狮

2020-07-02

合法字符串trim后，按照.分割成4个串，再分别转换成二进制，最后合并。这道题关键应该对于字符串是否合法的判断吧，比较能体现出一个人逻辑思维是否缜密



3