下面是几乎 50 道程序员电面题目的列表。这些问题可以用来考察任何程序员、开发者、软件工程师、测试和运营工程师，因为它们是基于程序设计的基础知 识的。但它们最适合程序员和开发者。顺便说一下，如果你是 Java 开发者，并且在寻找 Java 电面题目，去看看那个列表。本列表更加普遍，适用于所有的程 序员，包括 Python、Ruby、Perl 和 C#开发者。

1) 从哈希表，二叉树和链表中取元素需要多少时间？如果你有数百万记录呢？

哈希表需要 O(1)时间，二叉树需要 O(logN) (N 是树中节点数)，链表需要 O(N) (N 是链表中节点数）。如果数据结构工作正常（比如哈希表没有或只有相对少量冲突，二叉树是平衡的），数百万记录并不影响效率。如果工作不正常，那么效率会随着记录数上升而下降。

2) 覆盖(Overriding)和重载(Overloading)的区别是什么？ (detailed answer)

覆盖在运行时决定，重载是在编译时决定。并且覆盖和重载的机制不同，例如在 Java 中，重载方法的签名必须不同于原先方法的，但对于覆盖签名必须相同。

3) fork 一个进程和生成一个线程有什么区别？

当你 fork 一个进程时，新的进程将执行和父进程相同的代码，只是在不同的内存空间中。但当你在已有进程中生成一个线程时，它会生成一个新的代码执行路线，但共享同一个内存空间。

4) 什么是临界区？ (answer)

临界区是一段代码，十分重要，在多线程中同一时间只能被一个线程执行。可以用信号量或互斥量来保护临界区。在 Java 中你可以用 synchronized 关键字或 ReentrantLock 来保护临界区。

5) 值类型和引用类型有什么区别？ (answer)

值类型是更加优化的类型，总是不可变的(immutable)，例如 Java 原始的 int、long、double 和 float。引用类型指向一个对象，可能是可变的，也可能是不变的。你也可以说值类型指向一个值，引用类型指向一个对象。

6) 什么是在进程中的堆和栈？(detailed answer)

在同一个进程中，有两块不同的内存区域。以 Java 来说，栈用来存储原始值和指向对象的引用类型，但对象本身总是在堆中被创建。堆和栈的一个重要区别是，堆内存被所有线程共享，但每个线程有自己的栈。

7) 什么是版本控制？(answer)

版本控制是用来存储代码和管理代码库版本的软件，例如 SVN、CVS、Git、Perforce 和 ClearCase。它们在对比代码、审查代码和 从之前的稳定版本构造时十分高效。所有的专业开发都使用某种版本控制工具，否则你无法有效的管理代码，尤其是如果有 20 个开发者在同一个代码库上工作的时 候。版本控制工具在保持代码库一致性和处理代码冲突上扮演着十分重要的角色。

8) 什么是强类型程序设计语言？(answer)

在强类型语言中，编译器确保类型的正确性，例如你无法在 String 类型中存放数字，反之亦然。Java 是强类型语言，因此存在各种数据类型（如 int、float、String、char、boolean 等）。你只能将兼容的值存入相应的类型中。与此相反，弱类型语言不要求在编译时进行类型检 查，它们根据上下文处理值。Python 和 Perl 是两个常见的弱类型程序设计语言的例子，你可以将数字组成的字符串保存在数字类型中。

9) 可否描述一下有效(valid)的 XML 和格式正确(well-formed)的 XML 的区别？

格式正确的 XML 有根元素，所有标签都是正确关闭的，属性是正确定义的，它们的值正确地加上了引号。另一方面，有效的 XML 可以根据一个 XSD 文件或模式(schema)进行验证。所以一个 XML 可能是格式正确但不有效的（因为包含不被模式允许的标签）。

10) DOM 和 SAX 语法分析器有什么区别？(detailed answer)

DOM 语法分析器是驻留内存的，将整个 XML 文件装载到内存中，并创建一个 DOM 树进行语法分析。SAX 语法分析器是一个基于事件的语法分析器，所 以它根据收到的事件（如开始标签、结束标签、属性开始和属性结束）来对 XML 文档进行语法分析。根据他们的分析方法，DOM 语法分析器并不适用于大的 XML 文件，因为它会占用大量的内存空间，你的进程可能会耗尽内存。应该用 SAX 分析大的文件。对于小的文件，DOM 往往比 SAX 快很多。

11) 线程和进程的关系是什么？(detailed answer)

一个进程可以有多个线程，但一个线程总是属于唯一的进程。两个进程不能共享内存空间，除非它们有意通过共享内存进行进程间通信。但是同一进程的两个线程总是共享相同的内存。

12) 不可变(immutable)类是什么意思？(detailed answer)

一个类，如果在创建之后它的状态就不能被改变，那么他就是不可变的。例如 Java 中的 String。一旦你创建了一个 String，例如 “Java”，你就不能再改变它的内容。任何对这个字符串的改变（例如转换到大写、与另一个 String 连接）将创建一个新的对象。不可变的对象在并行程 序设计中很有用，因为它们可以在进程间被共享，不需要担心同步。事实上，整个函数是程序设计的模型都是在不可变对象上构建的。

13) 你为何要创建模拟(mock)对象？ (answer)

模拟对象在测试软件中一个独立的单元时很有用，事实上，存根(stub)和模拟都是创建自动化单元测试的有力工具。假设你在写一个显示货币兑换率的 程序，但没有一个可以连通的 URL，现在如果想测试你的代码，可以用模拟对象。在 Java 的世界中，有很多框架可以为你生成强大的模拟对象，例如 Mockito 和 PowerMock。

14) 什么是 SQL 注入？

SQL 注入是一种安全漏洞，它使得入侵者可以从系统中窃取数据。任何从用户那里得到输入并不加验证地创建 SQL 查询的系统都可能被 SQL 注入攻击。 在这样的系统中，入侵者可以输入 SQL 代码，而不是数据，来获取额外的数据。有很多用敏感信息（如用户 id、密码和个人信息）被人利用这种漏洞获取的实 例。 在 Java 中，你可以用 Prepared 语句来避免 SQL 注入。

15) 在 SQL 中，内连接(inner join)和左连接(left join)有什么区别？(answer)

在 SQL 中，主要有两种连接类型，内连接和外连接。外连接包括右外连接和左外连接。内连接和左连接的主要区别是，内连接中两个表都匹配的记录才被选 中，左连接中两个表都匹配的记录被选中，外加左表的所有记录都被选中。要留意包含“所有”的查询，它们往往要求左连接，例如写一个 SQL 查询来找所有的部 门和它们的雇员人数。如果你用内连接处理这个查询，你会漏掉没有人工作的空部门。

16) MVC 中的 V 代表什么，意味着什么？(answe）

在 MVC 模式中，V 是视图(View)。视图是用户看到的东西，比如网页。这是一个非常重要的 web 应用开发设计模式，它基于关注点分离原则，目的 是不同模块可以独立修改，不影响其他模块。在 Java 的世界中，有很多提供 MVC 模式的开源框架，例如 Struts 2 和 Spring MVC。顺便说一下，M 代表模型(Model)，C 代表控制器(Controller)。模型是实际的业务对象，例如用户、雇员、订单，控制器用来将

请求 分发给正确的处理单元。

17) 类和对象的区别是什么？ (detailed answer)

类是用来创建对象的设计图。一个类包括代码和行为，一个对象包括状态和行为。要创建一个对象，你必须创建一个表达对象结构的类。类还被用来在内存中映射对象，在 Java 中，JVM 替你完成这项工作。

18) 什么是疏耦合(loose-coupling)？

疏耦合是一种值得追求的软件特性，它使得对软件一个部分的修改不会影响到其他的部分。例如，在一个疏耦合的软件中，对 UI 布局的改变不应该影响后端的类结构。

19) 组合(composition)，聚合(aggregation)和关联(association)的区别是什么？ (detailed answer)

关联的意思是两个对象是相互联系的。组合是关联的一种形式，即一个对象由多个对象组成，但是它们必须共存，例如人体由各种器官组合而成，独立的器官不能生存，它们必须在身体内发挥作用。聚合也是关联的一种形式，表示对象的集合，例如城市是居民的聚合。

20) 接口和抽象类有什么区别？ (detailed answer)

这是所有程序员面试最经典的问题。接口是最纯粹的抽象形式，没有任何具体的东西。抽象类是一些抽象和具体事物的组合体。这个区别在不同语言中可能会不同，例如在 Java 中你可以扩展(extend)多个接口，但只能继承一个抽象类。更详细的讨论见于详细答案。

21) 什么是单元测试？(answer)

单元测试是测试独立单元（而不是整个应用程序）功能性的一种方法。在不同语言中，有很多工具可以做单元测试。例如在 Java 中，你可以用 JUnit 或 TestNG 来写单元测试。单元测试经常在构建时自动运行，或者在一个持续的环境（例如 Jenkins）中运行。

22) 你能否描述三种不同的在应用程序发布前对其进行测试的方式？

单元测试，集成测试，冒烟测试。单元测试用来测试独立的单元是否依照预期工作，集成测试用来测试已被测试过的独立单元能否共同工作，冒烟测试用来测试软件最常用的功能是否正常工作，例如在一个飞机订票网站中，你应该能订票，取消或更改航班。

23) 迭代和递归有什么区别？(detailed answer)

迭代通过循环来重复执行同一步骤，递归通过调用函数自身来做重复性工作。递归经常是复杂问题（例如汉诺塔、反转链表或反转字符串）的清晰简洁的解决 方案。递归的一个缺陷是深度，由于递归在栈中存储中间结果，你只能进行一定深度的递归，在那之后你的程序会因为 StackOverFlowError 而崩 溃。这就是在产品代码中优先使用迭代而不是递归的原因。

24) &和&&运算符的区别是什么？(detailed answer)

&是位运算符，&&是逻辑运算符。&和&&的一个区别是位运算符（&）可以被用于整型和 布尔类型，逻辑运算符（&&）只能被用于布尔类型变量。当你写 a & b 时，两个整型数的每一位都会进行与运算。当你写 a && b 时，第二个参数可能会也可能不会被执行，这也是它被称为短路运算符的原因，至少在 Java 中是这样的。我很喜欢这个问题，经常对初级开发者和毕业生问这 个问题。

25) 1 XOR 1 的结果是什么？

答案是 0，因为 XOR 在两个操作数（按位）不同时返回 1，相同时返回 0。例如 0 XOR 0 仍然是零，但 0 XOR 1 和 1 XOR 0 的结果是 1。

26) 如何得到一个整型数的最后一位？ (answer)

用取模运算符，数字 % 10 返回数字的最后一位。例如 2345 % 10 会返回 5，567 % 10 会返回 7。类似的，除运算符可以用来去掉数字的最后一位，例如 2345 / 10 的结果是 234，567 / 10 的结果是 56。这是值得了解的一个重要技巧，可以用来解决类似回文数、反转数的问题。

27) 什么是测试驱动开发？

测试驱动是一种常见的开发方法，在这种方法中，测试代码在功能代码之前编写。测试决定了程序的结构。测试驱动的纯粹主义者在写为应用写测试之前，不会写一行的应用代码。这能很大幅度地提高代码质量，经常被认为是巨星级开发者的品质。

28) 里氏替换原则(Liskov substitution principle, LSP)是什么？(answer)

里氏替换原则是鲍勃大叔称作 SOLID 的五条设计原则中的一条。里氏替换原则规定，所有的子类都能作为父类的代理(proxy)工作。例如，如果一 个方法需要父类对象作为输入，那么如果你提供一个子类对象，它也应该正常工作。任何不能替代父类的类都违反了里氏替换原则。这实际上是一个难以答出的问 题，如果你答出了，那么就会给面试官留下好的印象。

29) 什么是开闭(Open closed)设计原则？(answer)

开闭原则是 SOLID 中另一个重要的原则，它规定一个系统对扩展是开放的，但对修改是封闭的。意思是说，如果一个新的功能要被加入一个稳定的系统，那么你不需要碰已被测试过的现有代码，新的功能可以通过只添加新类来实现。

30) 二叉树和二叉查找树的区别是什么？

二叉查找树是有序的二叉树，所有节点（例如根节点）的左子树节点的值都小于或等于该节点的值，右子树节点的值都大于或等于该节点的值。它是一个重要的数据结构，可以用来表示有序的数据。

31) 你能否给出一个递归算法的实际例子？(example)

递归算法能适用在很多地方，例如与二叉树和链表相关的算法。几个与递归算法的例子包括反转字符串，计算斐波那契数列。其他的例子包括反转链表、树遍历以及快速排序。

32) 算法的时间复杂度是什么？

时间复杂度表示的是运行时间对输入量的比率。他表示一个算法处理一定量的输入需要多长时间。它是一个估计值，但足够表示输入量从十增长到一千万时，你的算法会有什么样的表现。

33) 链表和数组有哪些重要区别？(detailed answer)

链表和数组都是程序设计世界中重要的数据结构。它们间最明显的区别是，数组将元素存放在连续的地址中，链表将数据存放在内存中任意的位置。这使得链 表有巨大的扩展自己的灵活性，因为内存总是分散的。这种情况总是可能的：你无法创建一个数组来存放一百万个整数，但可以用链表来存放，因为空间是存在的， 只是不连续。其他的不同都是源于这项事实。例如，在数组中，如果你知道下标，可以用 O(1)的时间得到一个元素，但在链表中要花 O(n)的时间。更多不同 参见详细答案。

34) 在哈希表中处理冲突的方法都有哪些？

线性探测(linear probing)，二次哈希(double hashing)和链接(chaining)。在线性探测中，如果桶已经被占据了，那么函数会线性地检查下一个桶，直到找到一个空位。在链接中，多个元素可以存储在同一个桶中。

35) 正则表达式是什么意思？ (answer)

正则表达式是在文本型数据上进行模式匹配的方法。它是一种搜索的强有力方法，例如搜索长字符串中的某些字符，例如搜索一本书中是否含有某个单词。所 有主流程序设计语言都支持正则表达式，但是 Perl 正则表达式的能力是著名的。Java 的 java.util.regex 包也支持类似 Perl 的正则表达 式。你可以用正则表达式检查 email 地址是否有效，电话号码是否有效，邮政编码是否有效，甚至社会保险号(SSN)是否有效。正则表达式最简答的例子之 一是检查字符串是不是一个数。

36) 什么是无状态(stateless)系统？

无状态系统是不维护内部状态的系统。这种系统在任何时刻，对相同的输入都会给出相同的输出。编写优化一个无状态系统总是比较简单的，所以如果可能，你总是应该优先编写无状态系统。

37) 写一个 SQL 查询，在雇员表中查找第二高的工资。 (solution)

这是 SQL 面试的经典题目之一，尽管已经很老了，还是很有趣，并且可以追问很多问题来测试候选人的知识深度。你可以用相关或不相关的子查询来查找第 二高工资。如果你在用 SQL Server 或 MySQL，你也可以用类似 TOP 和 LIMIT 之类的关键字，前提是面试官允许。查找第二高工资的最简答方法是：这个查询首先查找最高工资，然后将它从列表中排除，再查找最高工资。很明显，第二次返回的是第二高工资。

38) 可否描述一下什么是关联的和不关联的子查询？(answer)

在关联的子查询中，内层查询依赖于外层查询，对外层查询的每一行执行。非关联的子查询不依赖于外层查询，可以独立执行。因此，前者慢，后者快。顺便说一下，关联的子查询有一些很棒的应用，其中包括在雇员表中查找第 N 高的工资，这在上一道 SQL 问题中也有提到。

39) 不用算术运算符，如何判定一个数是否是二的幂？(solution)

当你听到不能用算术运算符的限制时，应该立刻假定这是一道关于位运算的题。如果没有这条限制，那么你可以轻松地用取模和除运算符检查一个数是不是二的幂。用位运算符，有一个很巧妙的方法来完成任务。你可以用下面这段代码来检查一个数是不是二的幂

```
public static boolean powerOfTwo(int x) {
return (x & (x - 1)) == 0;
}
```

x & (x-1)是一个很棒的技巧，可以将最右边的为 1 的位设为 0。我是从《高效程序的奥秘》这本书中学到的。

40) 如何在 UNIX 上找到一个正在运行的 Java 进程？(command)

你可以组合使用 ps 和 grep 命令来查找 UNIX 机器上的任何进程。假设你的 Java 进程有名字，或者有任何可以用来匹配的文字，那么使用这个命令。

```
ps -ef | grep “myJavaApp”
```

ps -e 将列出所有的进程（所有用户的进程，不只是你的），ps -f 将显示所有细节，包括 PID。如果你想要深入调查或用 kill 命令杀死这个进程，你会需要 PID。

41) 如何在 UNIX 中寻找大的文件，例如 1GB 以上的文件？ (command)

你可以轻松地用 find 命令寻找大的文件，因为它提供依据大小寻找文件的选项。如果你的文件系统满了，你的 Java 进程因为没有空间而崩溃，那么就 使用这个命令。这个命令可以列出所有大于 1GB 的文件。

你可以很容易地改变大小，例如寻找所有 100MB 以上的文件，就用+100M。

```
find . – type f -size +1G -print
```

42) shell 脚本是什么？

shell 脚本是包含程序元素（例如 if 和 for 循环）的一组 shell 命令，它可以自动做一些重复的任务。例如，你可以写一个 shell 脚本来每天清理日志文件，为记录历史备份数据，以及其他家务活、版本发布、监视等等。

这个程序员电面问题列表到此为止了。你可能已经注意到了，只有 42 道题，但是标题说有 50 道，剩下的 8 道在哪？好吧，为了补偿这 8 道题，我在此分享 8 篇文章，你可以找到剩余的问题：

- 给程序员的 20 道字符串编程题(read here)
- 给软件开发者的 15 道数据结构与算法题(see here)
- 10 道所有开发者都应该指导的面试题(read more)
- 给 2 至 3 年经验程序员的 20 道核心 Java 问题(check here)
- 21 道 SQL 查询面试题与答案(read more)
- 给 Java 程序员的 23 道难题 (read here)
- 给程序员的 10 道 XML 面试题(see here)
- 来自 Java 面试的 50 道多线程和并发问题 (check here)
- 来自程序员面试的 20 道软件设计问题(read more)
- 18 道 Java 设计模式面试题(see here)

感谢你一直读到这里，如果你喜欢这篇文章，并觉得它对你的电话面试有帮助，请与朋友和同事分享。十分感谢所有提高面试题质量的建议。

扩展阅读：

Top 50 Programmer Phone Interview Questions with Answers

然后说说关于电面的建议：

- 准备好纸笔在旁边，不懂就写写画画；
- 说话不要停顿，讲不清楚的迅速带过；
- 要求远程 coding 的，记得提前熟悉线上编译环境（也有可能是白板）；
- **回答比较长的/把握不准的答案事先打印下来，电脑上开几个页面（面试官别打我；**
- 面试越靠前越基础，面试官 level 越低，但是考察越细致；反之；
- 基础要打扎实，口述比 onsite 难，因为对方看不到你的动作、表情等；
- 面试初想办法套（人）磁（肉）面试官；
- 摆一份个人简历在手边，防止出现描述不一致/记性不好的情况，如果是双面的，记得每一面单独摆一份；
- 不要让面试官在听你回答的时候听到点击鼠标和翻纸的声音（iPad 和 MacBook Pro 这种好东西我怎么会随便告诉你呢，嚯嚯嚯！）；
- 准备几个问题咨询面试官，这是必须也是礼貌；在碰到 HR 之前，别过分关心待遇，把咨询重心放在**核心问题**上；
- ..... (懒得写了。