



Java程序员 职场全攻略

——从小工到专家

吴亚峰 著

懵懂菜鸟如何才能成长为一名成功的开发人员？

如何在“血雨腥风”的求职大战中脱颖而出？

职场中只要技术过硬就可以吃遍天下吗？

是学招式重要，还是练内功重要？

掌握了哪些技能，才算是真正的高手？

.....

资深技术专家，娓娓道来。



 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

第 1 章 初窥门径——行业揭秘

IT，即 Information Technology，也有人恶搞地称其为挨踢。这是一个诞生没多少年的新兴产业，但却是个英雄和天才辈出的行业，这让行外人和行内人都十分赞叹。这个充满科技天才和财富的世界，究竟是什么样子的呢？本章就会带领各位读者揭开糊在 IT 行业上的这层纸。



提示 IT 含义很广泛，统指所有信息技术革命以来的计算机和电子通信技术。而本书中所指的 IT，一般只指计算机软件开发这一领域。

与读者同时开始 IT 职场之旅的，还有本书的两位主人公：

- 蔡佳娃，男，××理工大学计算机专业大三学生，做事比较踏实，小心谨慎，好钻研，但经常门路不对。
- 牛开复，男，毕业于××理工大学计算机专业，曾就业于 IBM 中国研发中心，目前在北京从事软件开发，业内高人。

1.1 IT 精英在中国的生存现状

IT 在中国的大发展不过近 30 年，但却是风起云涌、豪杰四起的 30 年。在中国的 IT 人中也不可否认地诞生了许多天才和富豪，但是这里比较关心的还是人数最多的大众 IT 人的生存现状，毕竟是千千万万的他们盖起了中国的 IT 大厦。

1.1.1 外行人眼中的 IT 人

不光是外行人，就连刚入行不久的新手，看 IT 这一行都是单纯的仰视和羡慕。面对这个新生而又成长极快并且正在迅速包围每个人生活方方面面的行业，很多人都是好奇与惊奇并存，羡慕与向往同在。总结一下，外行人看 IT 这个行业有如下几个特征：

- 高薪
- 年轻
- 高深
- 神奇
- 工作狂

升入大三的蔡佳娃，茫然间觉得一切都已迫在眉睫，大学四年已然过去了一半，是时候看看外面的世界了。蔡佳娃很了解一下两年后自己将要杀入的 IT 行业，于是门外汉的他便找到了已经混迹江湖大有所成的师兄牛开复。

1. 高薪

“牛开复师兄，你给句实话，咱们这行算不算高薪，到底挣多少？”

“呵呵，高薪，或许是你对这个行业的第一印象吧。我记得 2008 年中国收入最高的十大职

业调查中，IT 从业者的收入名列第二位，算是高的了。IT 这个行业先是被一道金色的光芒笼罩着，而且长期之内，这也是外行人向 IT 这边看时感受的第一束光。”

“那也就是真的高薪了？”

“也不能全靠调查，调查从本质上说是有些片面的，因为 360 行，行行出富豪，这种调查也只是一种总体上的、平均性质的。不过就算是个平均值，IT 这个行业还是能引起人们无尽的遐想。试问，谁会不自信到怀疑自己混不到平均值的水平呢？”

业内心声：

- 高薪是没错，但是 IT 是个贫富差距很大的行业，往往刚刚入门的程序员和高级顾问的收入差距会有几十倍。而中国的 IT 薪水又是呈两头小、中间大的梭子型分布，拿低薪和高薪的人少，大部分人都在中间徘徊。
- 在高薪的耀眼光芒下，谁也不知道有多少人毅然地奔向了 IT 的怀抱，但是如果粥还是那点粥，和尚越来越多的话，每个人分到的就会变少。
- 真正的 IT 精英收入依然较高，只是从菜鸟走向高薪和高手的道路因为同行者的增加而变得更加崎岖不平。不过这样的生存方式却很重要，只有这样严格遵循自然法则地发展，才使得 IT 这个行业在现在和未来大放异彩。

2. 年轻

“师兄，干这行的是不是都是年轻人哪？”

“何以见得啊？”

“我也不知道，就是一想到 IT 就觉得是年轻人干的事，可能是受一些网络或电视上媒体的影响吧，反正我还没有见过一个头发花白的开发人员呢。”

“那可不一定哦，IT 行业那些学术界的宗师和什么什么之父之类的可都是老先生呢。可能是我国的 IT 行业起步较晚吧，在我看来这一行的从业人员都比较年轻，不过你可以试试做一个中国年纪最大的开发人员。”

业内心声：

- 中国 IT 专业人员的年龄主要集中在 21~35 岁，其中 26~30 岁比例最高，占到四成；其次是 21~25 岁人群，略少与前者；31~35 岁居第三位，不足两成；剩下的不足半成的是其他年龄段的，大都是 35 岁以上的开发人员。
- IT 这个行业是一个充满激情的行业，所有人都是抱着迎难而上和求知若渴的心态在工作。所以，年轻行业和年轻的从业者，谁造就了谁，谁是因谁是果还真不好说清楚。年轻，或许更多指的还是一种心态。

3. 高深

“师兄啊，我以前，包括现在也是这么觉得，IT 这个行业实在是高深莫测。要让死气沉沉的机器按照人的意志来做事，IT 这工作真不是一般人就可以做得来的。”

“呵呵，这倒不假，干 IT 这一行对于头脑的要求还真不低，因为软件是个没有实体的东西，所有东西都必须是在脑海里的模型化存在，唯一面对的也只有洋洋洒洒的代码了。”

“就是啊，光面对那些看不到边的代码就够了，还要研究数学啦、算法啦等一般人几乎这辈子都不会深究的东西，光技术术语就一堆一堆的。”

“你说的没错，有时候我们也觉得要学的知识实在是太多了。不过你慢慢会发现，脑子就像是BT下载，用得越多就越好使。”

业内心声：

- 这倒不假，IT 的确是个聪明脑袋的聚集地，是个对个人能力要求比较高的行业。没有真本事会立刻被踢出队伍，IT 研究的是毫无趣味的代码和飘渺的算法，但是创造出来的却是极大方便人们的各种软件，这些软件功能强大，却简单易用。
- IT 人承担了机器语言和使用者的思想之间的翻译工作。其他职业要想培养出一个成手也许不会太难，而 IT 界，尤其是在竞争越来越激烈的形势下，想变成一个企业乐于聘用的合格开发人员，所要下的工夫绝对不仅仅是熬夜就可以达到的。

4. 神奇

“师兄，你知道我为什么学计算机这个专业吗？”

“不知道，脑袋被门挤了？”

“哪啊，我就是觉得 IT 这个行业很酷啊。好多新生名词不都是 IT 技术的发展催生出来的嘛。博客、贴吧、论坛……还有那么多的技术流行语。”

“很多技术术语都很深奥啊。”

“但是这种深奥不比薛定谔方程的那种不想懂也不想关心的深奥，这种深奥有种 fashion 的感觉，呵呵。”

“我懂你的意思啦，IT 给人的感觉很时尚是吧。”

“对喽。这就是我选择学计算机的原因！IT 世界就像个魔法世界，不断创造着让人眼前一亮的技术和流行词汇。”

业内心声：

- 的确是这样，IT 正以前所未有的速度改变着人们的生活，用神奇来说一点也不为过。光看中国网民的增长速度就会发现 IT 在互联网上吸引了多少人。
- 或许是只缘身在此山中吧，IT 人并不觉得这个行业有多时尚。其实也只有 IT 人才明白，神奇的背后可是无数个加班的日夜和费心的思考。当然，还有闪光的 idea。

5. 工作狂

“师兄，这个行业中的人是不是都是工作狂啊，干起活来不要命的？”

“是不是看新闻看的啊？”

“不光是新闻，其实也听其他人讲过，好多人因为不是工作狂而选择了离开这个行业呢。”

“你说得也不算全对，IT 有时候是需要工作玩命一些，因为脑力劳动有时候不能断，一断不好接，有时候封闭式开发一两个月不见天日。有时候在公司加班还不过瘾，还把工作带回家去做。这些都有可能发生，但是我们玩命更多的是源自激情。”

业内心声：

- 大部分 IT 人都会有一些工作狂，因为脑力劳动不比体力劳动，脑力劳动相对不容易疲劳。尤其再加上不少 IT 从业人员都爱较劲，所以工作热情一上来，工作拼点命也就比较常见了。
- 但是有时候工作狂也可能是因为主管交给员工的任务没法按时完成，或是领导交给主管的任务没法按时完成，所以这种拼命就很无奈了。

上述几个特征，大概就是外行人对 IT 人的印象吧。总的来说，这是一个说起来很有面子的行业，时髦、多金。同时，这个行业的人工作起来比较拼命。或许正是这些特征，吸引着一批又一批的年轻人投入其中，创造着自己的神话。

1.1.2 IT 行情分布

为什么要提一下中国的 IT 行情分布呢？因为很多有志于从事 IT 行业的人，对整个 IT 行业在中国的行情并不了解。单纯地认为只要是种子，在哪里都会发芽，却往往忽略了自己职业生涯的成长与目标。例如蔡佳娃同学就是这样。

“蔡佳娃，毕业了去哪里啊？”

“我只是想着如何自信满满地毕业，还真没想过该去哪。去哪不都一样吗？”

“怎么会一样呢？就算是一种花，也有它适宜生长的地方，什么地方开得艳，什么地方开得蔫，什么地方不会开花，什么地方年年开花还节节高。”

“师兄你是说我是那株花？”

“不光你是，整个 IT 行业也是啊。”

“我觉得堂堂一个 IT 产业，不该会有地区的差异和限制吧？”

“不仅有，而且是相当大啊。所以，在你还没有实力改变土质的时候，最好选择一块好的土壤。”

“那师兄你就给我讲讲去哪扎根吧！”

本节就来研究 IT 这朵花的生长习性，这里所指的行情分布，主要分为以下两点：

- IT 从业人员分布
- IT 服务性质

1. IT 从业人员分布

“先给你看看中国的 IT 从业人员分布表吧，看看中国的 IT 精英们都聚集在哪里研究问题呢。”
(见表 1-1)

“哇，分布太不平衡了。北京、广东和上海的 IT 从业人员几乎就是全国 IT 总人数的一半啊。”

“所以说嘛，中国的 IT 人才分布还是很不平衡的，而且就当前情况来看这种状况会持续很长一段时间啊。”

“人多能说明什么啊？”

“IT 最基本的单位就是人才，所以人才多的地方机会就多，技术的层次就会较高。而一些从业人员少的地方，也在一定程度上反映了那个地方的市场需求，所以机会不多，也就很少有机会接触新技术了。也就是说蛋糕如果很小，那就用不着刀子切了，一口就可以直接吞下去。”

“那选择地区就应该优先选择人多的地方喽？”

“也不尽然，不排除某一天其他某个地方会出现另一个‘硅谷’的可能哦。比如 IBM 头脑一

热把中国研究院迁到了×××，那可能就是另一个‘硅谷’呢。不过一般情况下，为了自己更好的发展，还是尽量去那些人才和技术比较集中的地方。”

表 1-1 IT从业人员分布表

省 市	IT 从业人员比例	省 市	IT 从业人员比例
北 京	22.4%	河北省	2.3%
广东省	15.8%	四川省	2.1%
上海市	11.5%	湖北省	2.0%
江苏省	7.3%	江西省	1.9%
浙江省	5.2%	天津市	1.8%
辽宁省	4.4%	河南省	1.6%
山东省	3.5%	黑龙江省	1.5%
福建省	3.5%	重庆市	1.3%
陕西省	2.6%	安徽省	1.1%
其他省市	8.2%		

由表 1-1 可知，中国的 IT 从业人员分布一般集中在东部沿海一带，而东部沿海一带又以北京、上海、广东三个地区所占的比重最大，所以这些地方也都是中国 IT 技术的前沿阵地，拥有最高级的开发人员，拥有全国最先进的技术。当然，也存在着全国最激烈的竞争。

2. IT 服务性质

“刚才我们谈了 IT 行业地区分布的差异，现在谈一谈服务性质的差异吧。”

“服务性质？”

“对啊，首先你要明确一个概念，IT 产业属于服务业，就像医疗、教育、金融一样。既然是服务业肯定就有服务对象，这里我们不会分得太细，只是将软件开发产业面向的客户群体做一个简单介绍。”

“啊，那师兄你说吧。”

“中国的 IT 产业，一大部分都是‘自产自销’，服务于国内的各个行业，比如电力、银行、钢铁等，我们姑且称之为传统的软件产业。”

“另外一种呢？”

“另外一种也是近几年发展迅速的运营方式也就是软件外包产业。软件外包充分利用了劳动力成本和全球化的优势，接受主要来自欧美、日韩等劳动力成本较高的公司的外包业务。”

“那中国比较知名的外包公司和地区大概是哪里啊？”

“提到中国的软件外包，不得不提大连，大连是中国最为出色的对外外包基地之一，由于地理位置等得天独厚的有利条件，使其在主要面向日本的 IT 公司外包业务上获得了很大的成功。最著名的软件外包公司东软就在大连。”

“软件外包的发展会很好吗？”

“我个人比较看好软件外包，因为一方面国家政策很扶持，另一方面我国的软件外包优势很明显，软件外包在未来几年肯定会有更好的发展。”

中国传统的软件开发产业也可以分为以下两种开发形式：

- 客户提出需求，在自己的公司做项目开发，项目开发和测试完成后，再拿到客户所在地进行部署调试。这种开发方式应该比较符合职场新手对软件开发的理解。
- 客户提出需求，自己派团队到客户所在地进行项目开发，并随时部署调试。这种开发方式工期短，效率高，反应速度较快，一般用在政府、银行等大型项目中。这种项目一般安全性要求也非常高，所以很多时候都是封闭式开发。

软件外包，就是接到项目后其中的一部分自己不做，承包给别人去做。软件外包中的发包方就是将项目承包出去的一方，是接包方的客户；接包方就是赚钱的一方，负责将分给自己的项目做好。软件外包产业是个非常有前景的产业，中国正在努力让自己成为一个全球性的软件外包基地，就像印度一样。

中国的软件外包主要分布在北京、广东、上海、辽东等地区，其中对日外包所占比例较大，其次是美国和欧洲。随着全球化步伐的加快和中国软件技术水平的提高，中国软件外包竞争力将会大大增强，中国正在成为全球数一数二的软件外包基地。

软件外包开始越来越独立出来，由国家政策的扶持和业内巨头的牵头，出现了很多像大连这样的外包产业基地，如苏州工业园区等。软件外包在聚集的同时，也在向 IT 中心城市的周边发展，因为软件外包本来就是劳动力成本差异而产生的，因此一些周边城市的成本优势就很明显了。

1.1.3 IT 语言平台

软件开发这个行业的武器就是编程语言，IT 发展到今天，产生了种类繁多的编程语言，有的古老却仍然实用，有的新生而激情四射，有的只是昙花一现就黯然消逝，有的则从诞生之日起就不断发展壮大。选择何种编程语言平台，便是摆在很多希望步入 IT 行业人的首道难题。

“蔡佳娃，说说看，你都学过什么编程语言啊？”

“不算少哦，我学过 C、C++、选修过 VB、下学期还会有 Java。”

“那你准备选哪种语言入行呢？”

“这倒没想过，不过只要学得多了都有好处吧？”

“那可不一定，你学得多我不反对，不过必须得有一门精通的语言，就像种了十盆花，总得有一盆是拿得出手的吧？”

“恩，那倒是，那师兄你给我介绍介绍当今的编程语言行情呗。”

“OK，先给你看看最新的编程语言排行榜。”（见图 1.1）

图 1-1 列出了 2009 年 9 月编程语言排行前 12 名的柱状分布图，可以看出 Java 虽然优势不是特别大，但还是顽强地占据开发语言老大哥的位子。其他比较强势的语言是 C、PHP、C++ 和 VB，这几门语言占据了所有编程语言的半壁江山还要多。

1. Java

Java 语言自 1995 年发布以来，由于其面向对象、跨平台和分布式的特性迅速风靡整个 IT 界。加之 Sun、Oracle、IBM 等行业巨头的大力支持，以及众多开发者为其设计的各种框架技术，使得 Java 技术涵盖了当今软件开发的几乎所有方面。其中，Java EE 和 Java ME 分别在企业级和移动开发上牢牢占据着霸主的地位。

另外，由于 Java 主要面向上层应用，运行在 Java 虚拟机上，所以无法对系统底层进行很强的

操作。因此底层系统的开发，如操作系统、51 单片机等，一般是不能采用 Java 技术的。

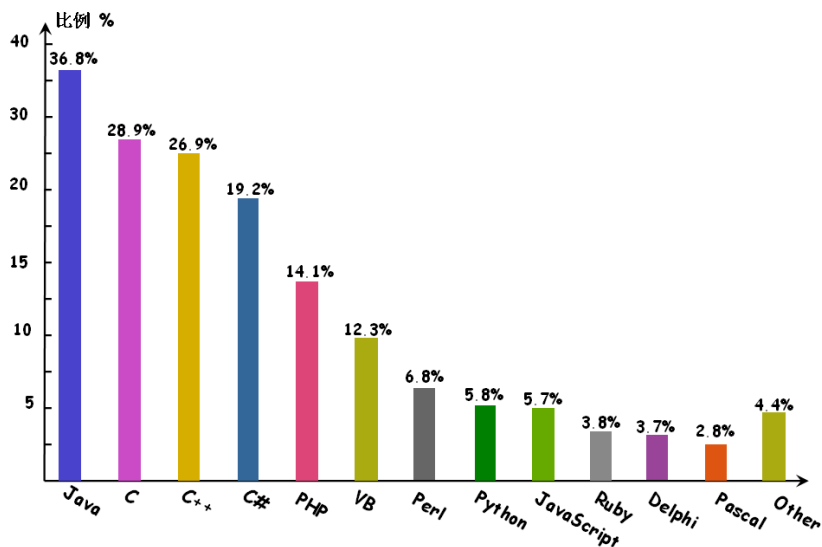


图 1-1 编程语言分布柱状图

2. C

C 语言历史悠久，功能强大，很多编程语言都或多或少地受到 C 语言的影响，发展到今天仍然可以傲视大部分后来者。C 语言是一种面向过程的结构化语言，由于其面向底层，编程灵活，效率高，广泛应用在嵌入式、操作系统等系统级别的开发中。

早期 C 语言也用来做上层软件的开发，近些年由于其他专门针对上层开发的语言平台（如 Java）的长足发展，C 语言逐渐淡出了上层软件的开发。

3. C++

C++ 衍生自 C 语言，但是发展到现在 C++ 和 C 语言已经独立开来，并且 C++ 编译器也可以容纳 C 的源代码。C++ 很少被用在 Web 级和企业级开发，更多地被应用在系统级开发的项目中，在大型游戏、设备驱动等方面有较大的优势。

但是由于 C++ 语法并没有彻底甩掉 C 语言的累赘，面向对象搞得不够彻底，使得 C++ 语言过于复杂，可靠性要略低于其他语言。

4. C#

C# 是微软开发的面向对象编程语言，C# 有许多优于 C++ 的特性，比起 C++，C# 在面向对象方面做得更好。C# 用作 Web 开发比 C++ 好，而底层开发却远不如 C++，C# 需要运行在 .NET Framework 之下，所以有一定的跨平台性。



提示 .NET Framework 实际上就是个虚拟机，但很遗憾的是 .NET Framework 只有 Windows 系列操作系统中才有，这也在很大程度上影响了 C# 的覆盖面。

5. PHP

PHP 是一种内嵌在 HTML 中的服务器端脚本语言，它是开放源代码的。经过历代开发者的辛

勤劳劳动，PHP 从 1.0 版的只有一个简单的网站计数器和留言本，发展到 PHP 5 强大的面向对象功能。在开源项目如火如荼的发展形势下，PHP 应该还会吸引大批的开发者参与进来。

不过 PHP 由于一些线程安全问题、缺少标准框架和其他商业原因等（不要认为开源的就是彻底免费的），PHP 在开发一些大型复杂的项目的时候就有些捉襟见肘，因此 PHP 比较适合于中小型的项目开发。

6. (Visual) Basic

Basis 最初是为初学者设计的一门语言，后来被计算机厂商（尤其是微软）不断改进。Windows 操作系统问世后，Visual Basic 逐渐流行，Visual Basic 最大的特点就是易用，可以方便地连接数据库，快速建立 Windows 应用程序和企业级程序。

同样是由于 Visual Basic 的易用性，很多人认为不应该将其作为初学者的入门语言，不利于初学者学到基础的编程模式和结构及养成良好的编程习惯。而且用 Visual Basic 开发的程序只能运行在微软的 Windows 平台之下，这也在一定程度上制约了其用武之地。

7. Perl

Perl 是一种具有动态特性的脚本语言，同时支持泛型变量等灵活的特性。由于 Perl 借鉴了其编程语言的很多语法，使得 Perl 比较容易学习。Perl 主要应用在 UNIX 平台下的程序开发和 PHP 的开发中。具有一定的跨平台性能。

8. Python

Python 是一门比较年轻的语言，在 2004~2005 年非常流行，随后发展平稳。其最大的特点就是可以和 Java、C++ 等语言很好地结合在一起，而且只要添加某种模块，就可以实现相应的功能。Python 主要应用于多媒体处理、网络编程等方面。

9. JavaScript

JavaScript 不同于 Perl 等脚本语言，JavaScript 是运行在客户端的脚本语言。其简单易学，但是要想用好必须下大工夫学习。尽管安全性差，JavaScript 还是靠着其跨平台性、灵活性高等特点成为最受欢迎的脚本语言，尤其是在如火如荼的 AJAX 技术渐渐成为大众新宠之后。



提示 很多初学者容易混淆 JavaScript 与 Java，其实二者之间主要的联系就是名称当中有四个英文字母相同，其他的如应用领域、运行平台等都有很大的区别。Java 可以开发从单片机嵌入式程序到大型企业级应用，而 JavaScript 主要是开发嵌入在浏览器中运行的应用程序。

10. Ruby

Ruby 和 Python 以及 Perl 比较类似，最大的不同之处在于 Ruby 里所有都是对象（而在 Java 里字面常量被封装之前并不是对象），而且 Ruby 比后两者都年轻。Ruby 有一个基于 MVC 模式的框架 ROR（Ruby On Rails），在解决中小型应用时快速简洁。

Ruby 由日本人发明，所以网络上的资源并不是很多，关于 Ruby 是否能流行于大型商务应用，笔者持观望态度。

11. Delphi

Delphi 是由 Borland 公司开发的，它并不是一门语言，更多的是个开发环境，主要使用 Object Pascal 语言，很多人熟悉的“熊猫烧香”就是用 Delphi 开发的。除了开发 Windows 下的应用程序，Delphi 还可以应用在 Linux 平台下。



提示 Linux 平台下的 Delphi 不称之为 Delphi，而称为 Kylix。不过由于 Windows 与 Linux 操作系统平台所提供的接口不同，并不是所有用 Delphi 开发的软件都可以无缝迁移到 Kylix 中。

12. Pascal

同 Basic 一样，Pascal 一开始也是为教学而设计的。因此 Pascal 是一门很好的入门语言，用于数值计算也非常适合，也是很多计算机类大赛的参赛语言，同时很多名牌大学也是将其作为入门教学语言。Pascal 语言产生了很多版本，Delphi 采用的就是其中的 Object Pascal。

“怎么样，蔡佳娃？看了这么多编程语言的介绍，心里有没有什么打算啦？”

“师兄，越看越迷糊啦！”

“没关系，可以好好想想，我说的可能不全，你可以在网上再搜一搜资料。”

“OK！”

选择一门编程语言，只是入门的途径。过分依赖编程语言，只会让自己成为代码高手，而不是开发大牛，要知道编程语言只是一种工具，更重要的是编程思想。

1.1.4 你说我容易吗

前面介绍了外行人眼中 IT 人的印象，其实家家有本难念的经，在貌似风光的职业背后，IT 人也有 IT 人的辛酸，也有一些不足为外人道的苦衷。

下面向读者介绍几个 IT 人不容易的方面：

- 水涨船高的薪水
- 脑力负荷重
- 技术更新快
- 压力大

“蔡佳娃，上回你讲的是你对 IT 这个行业的看法，这回换我这个行内人给你吐吐我们的苦水了。”

“是吗？那我得好好听听哪！”

1. 水涨船高的薪水

“就像我之前说的一样，IT 从业人员大都分布在东部沿海地区，而这些地区恰恰也是中国物价比较高的地区。所以就薪水来说，北京的年薪十万和石家庄的年薪十万是大大不同的概念。”

“听你这么一说，果然有道理。”

“不仅如此呢，除去物价，必须考虑到现在中国越炒越高的房地产，想想中国房价最贵的地方，温州、上海、北京、杭州、深圳、广州……那些 IT 中心城市，哪个房价不是排在前头啊。只能说，不幸，每个 IT 精英所向往的工作地同时也是中国最寸土寸金的地方。”

“师兄，我错了，原来 IT 人也不容易啊！”

业内心声：薪高一尺，价高一丈，所以要想在 IT 界精英的聚集地安个家，对于每个 IT 人都是个不小的难题。但是，一个真正的 IT 人凭借着自己的努力，总是可以实现这个目标的。当然需要的时间就只能因人而异了。

2. 脑力负荷重

“还有啊，虽然 IT 人挣高薪挣得有些假，可是工作负荷重却是毫无疑问的。”

“此话怎讲呢？”

“IT 几乎是个纯脑力劳动的产业，对于细节的把握非常重要，因为一个小小的编程错误可以造成数百万甚至上千万的损失。本来劳动强度就够大，还要不时加班熬夜，很多人步入 IT 行业一两年后，由于太累扛不住，转行的人也不少。”

“哎，的确是这样哈，IT 人赚的每一分钱都是智慧和死亡脑细胞的结晶啊。”

业内心声：除了 IT，大概找不出第二个对智力依赖如此高的服务行业了，严谨求实的就是这个行业的标准。再加上熬夜和加班，开发人员的脑子承受着不一般的负荷。开发人员一边为算法的优化费尽心思，一边又在为找 bug 排除故障抓破头皮。

3. 技术更新快

“蔡佳娃，你觉得我们 IT 行业发展快不快啊？”

“那是相当快啊！日新月异都不行，得是‘分’新‘时’异。”

“发展快对谁都好，就是对我们开发人员不好，这就意味着我们必须抓紧时间学习新技术，在 IT 行业，新技术就是逆水，不学则退啊。好多公司每年都投入很多财力物力来搞培训。”

大家都知道庄子在《庄子·养生主》中曾经说过：“吾生也有涯，而知也无涯”，但是大家几乎全都忘了后面几句：“以有涯随无涯，殆已；已而为知者，殆而已矣”。全文的意思是：我们的生命是有限的，而知识是无穷尽的，用我们有限的生命去探求无限的知识，肯定不会有好结果的。

然而 IT 这个光鲜亮丽的行业却必须是一个需要不断学习的行业。IT 仅仅在其诞生的几十年中就已经发展到如日中天的地步，靠得不正是迅猛的技术更新速度！光看硬件产业，在著名的摩尔定律的带领下，计算机配件的性能和价格变脸般地更新着每个人的生活节奏。

所以如果你在满负荷的工作之后没有抓紧时间学习新技术，那么结局大概就像《多收了三五斗》里那样，只是多混了三五年经验，结果反倒贬值了。

4. 压力大

“说到这，我如果跟你说 IT 行业的压力很大，你还有意见吗？”

“呵呵，绝对没有了，看来 IT 这个东西，不是人人都能扛下去的啊。”

“是啊，不过如果你能扛下去，做到了高层，或者是技术元老，站到技术的最前沿，你就真算是修成正果了啊！”

“师兄。希望你赶快变成高层啊！”

“恩那！呵呵。”

业内心声：概括来讲，IT 人享受着平均高薪，普遍高压，追赶着先进技术，做着脑力劳动。这些可能不是每个 IT 人现在正在做的，但却是每个 IT 人曾经做过的。总而言之，IT 是个勇敢者的游戏，没胆量的人是玩不到头的。

1.1.5 我挨踢我骄傲

尽管 IT 人存在着前面介绍的种种难处，每年还是有很多人涌入 IT 这个职场淘金，而已在 IT 很多年的老手也很少会去转行。既然是这样，那么干 IT 这一行绝对还是有很多让人留连忘返之处的。

- 外人的眼光
- 高成就感

1. 外人的眼光

一项工作体不体面都是外人说了算，很多外行人都一般地认为从事软件开发的人脑子都特别好使，这让很多开发人员听了，就算工作很苦，压力很大，还是觉得很值。

诚然，现在一些 IT 公司员工猝死的事件也让外界对这个行业的玩命程度肃然起敬，但是 IT 这个流行职业还是以其超赞的智商劳动和前沿的技术手段惊喜着每个人。累是累，但是 IT 人吃的都是智商做的饭，香！

2. 高成就感

用自己的聪明才智赚钱，肯定是很欣慰的，况且从事的还是对很多人讲都讲不明白的高新技术。身为 IT 人，看着闯入人们生活的种种 IT 产品，那种通晓内幕的沾沾自喜是很难有其他事物能够比拟的。

设想一下，你是淘宝网后台系统的开发人员，你看到别人在淘宝上买东西，你会很自豪。因为你通晓这个用来买卖东西平台的来龙去脉，了解整个业务流程，甚至还知道一些别人不知道的 bug 或窍门。此时，再看着他在电脑上一步一步按着你定下的规矩操作，内心的窃喜是不言而喻的。

就算很少有人听你讲解这其中的奥妙，因为在他们看来这不过只是“茴香豆”的四种写法，但是身为 IT 人，感受那种真理掌握在少数人手里的感觉，妙不可言。

1.2 当今主流公司的企业文化

除了高校和科研院所，IT 人才基本上都是集中在各个公司的。而各 IT 公司因为文化背景、行业领域、公司规模等方面的差异，也存在着不同的企业文化。IT 人在入行之前研究清楚什么样的公司唱什么样的歌，根据自己特点选择合适的公司，防止自己“入对了行却选错了铺子”是很有必要的。

1.2.1 欧美企业的特色文化

计算机技术起源于美国，当然欧洲也贡献了大量的优秀数学家和计算机学者。所以欧美（当然也包括日本）的 IT 技术是全球领先的，欧美公司大概是每个 IT 精英都想去一展身手的地方吧。因此，欧美公司的特色文化必须先搞清楚。

“蔡佳娃，上回让你看了编程语言的排行，现在做决定了吗？”

“不急嘛，师兄。你再给我多介绍介绍呗。”

“OK，今天我来跟你说说现如今各个公司的企业文化。”

“企业文化？有必要吗？甭管什么公司直接进去干不就行了吗？”

“哎，你总是逼我鄙视你的无知啊。不同的公司当然有不同的文化，听我讲完你就大概明白了。”

“呵呵，那师兄你赶紧给我讲讲吧。”

“先说说欧美公司的企业文化，欧美公司的技术含量是最高的，而且欧美公司是每个 IT 精英都最希望去的理想企业。”

“这是为什么呢？高薪吧？”

“不全是，主要是欧美公司比较人性化，工作氛围十分自由开放，而且上下级各个阶层交流都非常直接，有些地方还会实行弹性工作制，就是尽最大限度让员工感到安逸，进而很自愿，很自觉，很自发地努力工作，证明自我。”

“的确啊，看来欧美公司的确是我们的天堂啊！”

“不过，一分钱一分货，没有真本事，欧美公司的门槛我们是很难挤进去的哦！”

“千真万确，不过我很期待自己能有一天得偿所愿，呵呵。”

欧美公司（主要是美国的公司）比较根本的原则就是以人为本。欧美公司认为一个企业最重要的军事力量是人才，而人才是需要被尊重、保护的，所以那些企业鼓励员工学习培训，努力让员工感到适应，从而心无旁骛。

欧美公司重视人才，也体现在其招聘人才、构建最适合工作的环境、员工福利方面，不过就像故事里讲的那样，多大饭量多大碗，虽然欧美公司招聘人才不拘一格，但是想要进去，恐怕还是需要自己头上顶上了“精英”的光环才行。

这里需要提一下德国的 IT 企业文化，德国素来是科技实力比较高，学术氛围比较浓厚的，高斯、哥德巴赫，莱布尼茨等伟大的头脑都来自德国。德国的企业严谨踏实，纪律严格，跟中国比起来可以说毫无人情味儿，不过也不完全是坏处，只需要搞好技术，可以不必研究烦琐的人际关系，这也是很多 IT 人所不擅长的一项。

最后，以微软为例，简要说明一下欧美 IT 企业的基本团队模式，如图 1-2 所示。

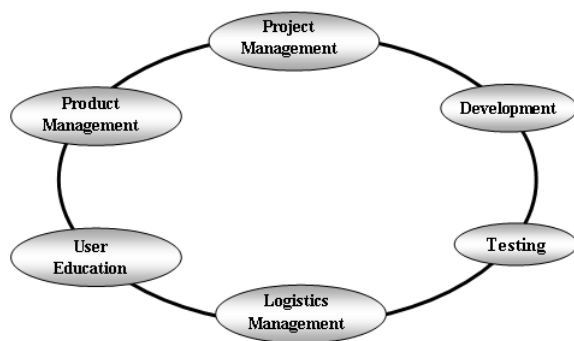


图 1-2 微软项目团队模式图

首先从图 1-2 中可以看出，欧美企业内部的各个团队角色是平等的，没有森严的等级制度。

下面来介绍一下这个团队模式中的各个角色及其职能。

- **Project Management:** 项目经理，负责整个项目的总筹，掌握资金、人员等。
- **Development:** 开发人员，负责按照项目进行项目开发。
- **Testing:** 测试人员，和开发人员合作，进行项目测试。
- **Logistics Management:** 质量管理，项目测试无误后，内部或外部试用，验证项目软件的可用性及易用性。同时，也包括其他一些与项目相关的后勤工作。
- **User Education:** 用户培训，负责培训用户项目软件的使用法。
- **Product Management:** 产品经理，他是开发人员和用户之间的纽带，综合用户的需求，将项目的设计方案交给开发人员，或者将项目人员的观点反馈给用户。
- 另外微软项目管理主要是协调三个因素的平衡：
- **Resources**（资源）
- **Features**（功能特性）
- **Schedule**（时间表）

上述三个因素也称为黄金三角，如图 1-3 所示。

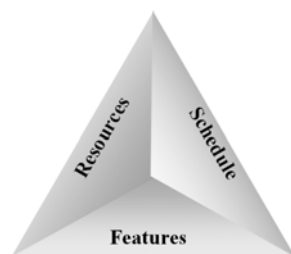


图 1-3 项目管理的黄金三角

其中资源主要包括人力和财力，在实际开发中需要根据客户的要求，将 3 个方面的因素进行调整。比如客户要求进度，那么在功能实现上可能需要妥协一些，资源要求也会比较高。

1.2.2 日韩企业的工作模式

日本和韩国的 IT 企业在中国也有不小的规模，尤其是日本和韩国（主要是日本）的软件外包业务占中国的软件外包业务总量的比重很大，所以那些日韩外包的接包企业也或多或少地受到了日韩企业文化的影响。

“刚才我们讨论了欧美企业的文化，现在我们来讲讲日本和韩国的企业文化。”

“日本和韩国的企业也很不错吧？应该比国内的好吧？”

“别急，听我说完就知道了。每个人的看法都不一样。下面我只谈谈我的看法。”

“恩，你说吧，我洗耳恭听。”

“我听我一个朋友说日韩企业的管理方式就是吸血鬼式的管理，千辛万苦地榨取员工的每一滴血，千方百计让员工拼命。在那里，不加班是不合群的表现。”

“啊，这么残酷啊。”

“这就是日韩企业的最大特色，不过这并不能算是坏处，你想想看二战过后日本和韩国为何经过几十年的发展就迅速成长为经济发达的国家，就是靠着这种玩命的干劲。你肯定听说过日本是

世界上工作压力最大的国家吧。”

“恩，不过这样应该还不错吧，至少可以让公司和个人都得到发展呢。”

“光是这样当然还可以，不过日本企业的等级制度比较严格，和中国的企业有些类似，所以办个什么事情比较麻烦。”

“恩，这样一来还不如咱中国的公司好呢。至少是为自己人拼命。”

“呵呵，你看，愤青了吧。其实日韩的 IT 企业主要讲究的就是团结，跟欧美公司最大的不同就是日韩企业讲究团体实力，而欧美公司更加侧重打天才战术。就好比美国的 NBA 或其他体育项目喜欢打明星战术，而不像日韩的团队体育精神。”

其实故事里关于日本 IT 公司有一个方面没有提到，那就是日本公司对员工的福利特别好，也算是对员工玩命工作的一种补偿吧。

韩国的 IT 公司和日本差不多，不过韩国的公司对员工的企业文化培养比较重视，真正工作之前会接受比较全面的培训，再有就是韩国 IT 公司的纪律也是非常严格的，这一点和日本公司差不多，所以有时会感觉很受限制。

其次，韩国的升职制度也比较成问题，所以在韩企升到高层比较困难，不过韩国公司的坚强干劲还是很值得大家学习的。

1.2.3 中资企业的传统特色

既然是中国的 IT 行业，存在最多的公司必然是中国自己的公司。中国的 IT 公司在 IT 大发展的三十年间逐步改进，受到欧美和日韩等国家的企业文化影响，再结合中国传统企业文化的积淀，逐步形成了比较中庸的企业文化，这也比较符合中国传统的儒家文化。

“讲到了最后，终于该提提我们自己国家的 IT 企业文化了。”

“就是啊，我也纳闷呢。为什么不先说我们国家的呢？”

“我这么做是有原因的。欧美和日韩的企业文化差异比较大，而中国的企业文化比较中庸，前两者的特点兼而有之，所以我们放到最后说。”

“哦，那师兄你说说怎么个中庸之道吧？”

“虽然中国的官本位文化比较浓厚，但是在 IT 这个比较前卫的行业，中国的公司受欧美的影响还是比较大的，还是有一些以人为本的特征的。”

“是吗？这样可太好了。”

“我有一个朋友就任职于一家中国的 IT 企业。他们那里就很有人情味儿，上级和下级的关系都很好，纪律也不是特别严格，也很少加班，但是很多事情还是领导说了算。”

“这也不错哈，工作环境应该比日韩企业要好一些。”

“其实有些地方也和日韩企业比较相像，比如也会有很多篮球、足球之类的社团，公司也会找时间组织员工去旅游啊什么的。”

“哦，看来中国的 IT 企业的确比较中庸，比不上欧美公司的激情，也没有日韩公司的严格。”

跟欧美其他国家不一样，我国的 IT 发展时间不长，而 IT 在短短的历史中诞生了大量的技术。因此大量技术更类似是“涌入”的，再加上欧美等其他国家公司的进入，对我国的 IT 企业造成了

多方面的影响，所以显得中国的 IT 企业看起来就像是欧美和本国传统企业的混合体。

同样是由于上述原因，中国的中小型 IT 企业远多于大型 IT 企业，这也就使得中国 IT 行业中的团队模式一般规模不会很大，分工也不是很细致，比较类似于欧美的自由模式，但又很像日韩的官僚等级，其实官僚主义并不是什么特征，任何比较大型称得上“帝国”的公司都会滋生这种东西，比如 IBM。

中国的 IT 企业文化还在不断地探索自己的道路，既接受着外国企业的文化冲击，又摸索着自己的道路。国外的文化不一定全有用，不可单纯地搞“拿来主义”，中国的企业需要自己的特色。比如中国的员工私下关系都很好，而外国的企业很大程度上都是为了工作才交际。

1.2.4 两种不同的软件外包方式

在 1.2.2 节中，本书向读者介绍了中国软件外包企业的行情。软件外包产业将会是中国软件出口的一项重头武器，所以本节将继续向读者介绍现今存在于中国的两种不同的软件外包方式：人员流动和项目流动。

“蔡娃娃啊，还记得上次我们说到的 IT 软件外包产业吗？”

“记得记得，主要面向日韩欧美的嘛。”

“恩，不错。今天我们就来说说中国软件外包产业的两种不同方式。”

“这还有分别啊？不就是人家让干啥就干啥呗，完事了给钱就行了吗？”

“此言差矣，哪有这么简单啊，老是这么简单地想问题迟早会吃亏的。中国的软件外包产业分为两种方式：人员流动和项目流动。”

“那师兄你给我讲讲吧。”

1. 人员流动

人员流动，即发包方向接包方要求开发人员的租用，接包方将开发人员送到发包方进行工作，实际上就是外包的内容是人。被租出去的开发人员便会去发包方公司上班，而薪水和其他福利政策则由接包方负责发放和实施。

外包出去的人员并不一定都是开发人员，很多公司并不缺少开发人员，而是缺少软件测试人员，所以外包软件测试人员的也很多。

“我们先谈人员流动，人员流动其实说白了就是‘人口贩子’。比如说 A 公司打算开发一个软件项目，需要外包，但它不希望把项目拿给人家去做，所以它与 B 公司签约，从 B 公司借来一些开发人员，让他们参与到开发的过程中。”

“那那些被借来的开发人员到底算是哪边的人呢？”

“可以说生是 A 公司的人，死是 B 公司的鬼吧，呵呵。也就是说上班去 A 公司，做的工作也是在为 A 公司创造财富，但是薪水由 B 公司发，出了什么合同或者福利方面的问题也是去找 B 公司，从这一方面也说明，IT 人才的流动性是很大的。”

“哦，这样也很不错啊，有种双重间谍的味道，呵呵。”

“不错个鬼啊，就是因为有了这层关系，给很多假冒的猎头公司或者经营外包业务的公司带来了一些可乘之机，用来行骗。”

“行骗？那是怎么做得啊？”

“你比如说 B 公司招聘过来一些人，‘贩卖’到 A 公司去，而 A 公司自然不会为那些人的薪水、医保、保险等方面操心，而 B 公司又总是闪烁其词，一般会发发工资，但是其他的就不管了，这样员工一旦出了什么事情，A 公司和 B 公司双方踢皮球，最后受苦的还是员工本身。”

“哎，看来做什么都不能太急，必须研究无误了才能做决定，就像签合同。”

“是啊，‘You can never be too careful.’”

中国的软件外包企业中采用这种外包方式的还有很多，如比较出名的如软通动力、文思创新等。这种情况下一般发包方（即故事中的 A 公司）是一家规模比较大的公司，当其遇到比较大的项目时便会找像故事里 B 公司这样的公司来解决人员缺口的问题。

接到项目再去找人开发，这是一种很常见的公司运营模式，很多大的公司如 IBM 的 CSDL 就是这样，正式员工只有 50%，其他大部分都是“贩卖”过来的。所以被贩卖过去后如果表现上佳，还会被发包方聘为正式员工，那样就算是“从良”了。

人员流动是针对很多公司的这种节约成本的运营模式而产生的。较项目流动还是有一定先进性的，不过正像故事里说的那样，再遇到人员流动的外包情况时，一定要弄清楚自己的位置，保证自己出了什么事情都有人来管，切勿上当受骗。

2. 项目流动

项目流动，即发包方直接把项目的供求说明告知承包方，承包方收到后组织自己的开发人员（这时候的主旨人员就有可能用到人员流动）进行项目开发，然后在完成开发和测试后，将项目交给发包方。整个过程双方只有项目的交流，并没有人事上的走动。

“好了，介绍完人员流动，接下来再讲一讲项目流动。”

“恩，是不是 A 公司把项目交给 B 公司做，然后 A 再把钱给 B 公司啊？”

“很正确，这个应该比较符合你们对软件外包的基本想法吧？”

“恩，这个听起来就很像外包了。不过我觉得这种方式应该没有人员流动好吧？”

“从某一方面上讲，大概是这样，因为如果作为开发人员被外包到别的公司，你就会学到很多在原公司所没有的东西，同时也有可能为自己的职业生涯创造另一条路。我有一个朋友刚刚大学毕业居然在应聘的时候打败了有三年工作经验的一名老手，而那名老手之前就是就业于主营这种业务的外包公司的。”

“恩，师兄所言极是，我想说的就是这个意思！”

“不过不能一概而论，有些人就是喜欢安定，喜欢重复熟悉的事情。而项目流动方式就可以提供这些保障，而且项目流动方式并不是一条不好的职业生涯。”

项目流动是中国软件外包的另一种运作方式，这种外包目前在对日韩的一些外包业务上还是很流行，比如大连的一些外包公司都是采用的这种方式。

不过这种外包对开发人员的前途一般不好，因为有的时候发包方或者承包方已经把要开发的项目做了十分精细的分工，交到开发人员手里，开发人员只需要按照特别详细的设计做填空题就行了，设计类，编写函数，一切都是制定好的规则，有时会精确到把参数和返回值都规定好了。

所以在这种方式的开发中，开发人员并不能对项目有一个整体的把握，并且也失去了思考的机会，脑子也就停止了高层的运转。这样时间长了，在思维等其他方面自然也就落后于其他同行，不过这种方式倒是很适合初入行的新手锻炼磨合。

1.2.5 加入什么样的公司

“怎么样，蔡佳娃？听了这么多介绍，心里有点谱了吧？”

“恩，听师兄你这么一说，我想了想，还是优先要追求一下欧美的 IT 公司。追不到也没关系，至少知道自己不行了，可以继续再追，呵呵。”

“呵呵，精神可嘉啊。其实就算追不到，可以先去国企历练历练，因为中国的 IT 一直都是以欧美为模板发展的。”

“对，我就是这么个打算，最好不要去那些日韩的企业，虽然那些地方福利好，精神足，干劲强，不过我想还是让自己优先学习些新的理念和技术，最好还是少做些拼命的工作，毕竟身体是革命的第一本钱嘛。”

“你很会为自己着想啊。不过你的想法也没有坏处，如果我建议你的话，大概也是这么个策略，毕竟是搞 IT 的，就得向技术最先进的方向看齐嘛。”

“恩，欧美的技术先进，混进去至少能蹭个脸熟呢。”

故事中的建议内容大概只能作为参考，毕竟随着全球化的趋势越来越迅速，欧美、日韩和中国的 IT 企业都在慢慢地融合和交流。很多企业虽然还保留着自己的传统，但都在互相取长补短。真正的 IT 行业该走向何方，或许将由正在阅读本书的读者来决定。

不过请读者谨记，光嘴上说去哪是没用的，没有真本事是哪里都不能去的。有才能走遍天下，无才是寸步难行的。

1.3 散兵游勇还是团队作战

真正的软件开发行业并不像一些人想的那样，所有的项目由很多团队开发，也不像另一些人想的那样，全靠一个天才创造。两种情况都是存在的，关键是要在竞争如此激烈的 IT 行业做好这两种人，让自己有备无患。

1.3.1 哪样多一些

“师兄，我要问问你啊，要是真正进到了一家 IT 公司，一般自己干的时间多还是和团队集体干活的时间多啊？”

“你是想问团队合作开发的时候多还是自己独立的时候多吧？”

“恩，我就是这个意思，想知道自己以后要是真的进了公司，对哪方面要求高些。”

“这个是完全说不准的，而且是完全没有必要说准的，你考虑的太多啦，蔡佳娃。”

“那我总得做好准备吧？”

“你应该是全做准备。这两种方式你都要会，都能做得来，这样才行。”

“哦，我还以为做好一方面就行了呢。”

“不对，谁也说不好你会用到哪个。IT 这个行业的变数比较大。你加入的公司可能因为你跳槽等缘故改变，你所在的公司接受的项目规模也总在变，所以你必须两种情况都应付得来。”

“那至少哪个比较多，哪个比较少呢？”

“哎，你还是没有听明白，没有人去统计这个的。可以这么说，欧美公司（包括面向欧美的软件外包公司）因为接手的项目比较大，所以需要团队的开发会多一些；而中国的中小型 IT 公司比较多，它们要求个人的单独作战能力必须要强一些。”

前一小节介绍过，中国的中小型 IT 企业比较多，团队规模一般比较小，所接到的项目不可能很大，所以更多的情况是会把项目分成很大几块，然后交给团队开发，因为项目分割得很开，所以团队中每个成员所做的工作更多地像是一个独立的项目。

中国的一些大公司也有自己独特的经营模式和理念，不过在团队模式上相比欧美还是有所欠缺的，比较突出的一个就是中国的团队模式一般比较随意，分工比较模糊，规模也不太确定，不像欧美公司那样分工明细。

刚刚走出校门的大学生，自己单干的能力或多或少肯定是有，团队意识一般就很少有人有所体会。所以对这方面还是有所顾虑的，就像故事中的主人公，其实团队开发并不是什么高妙的东西，当项目经理把任务分配给每个人的时候，你接下来要做的就是自己来做，和自己开发是没有太大区别的。

1.3.2 团队和单兵

“师兄啊，团队协作的时候应该怎么做才好呢，究竟团队意识是什么啊？”

“团队意识这个概念很多时候都被放大了。其实你只要在团队中做对自己最有利的东西，同时也做对团队最有利的东西就好。”

“那自己单兵作战的能力呢？”

“不管你将来面临什么样的工作，这个能力是你必须要有的，单兵作战能力对整个项目的统筹和把握要求较高，对整个项目从需求分析到测试都必须全局关注和执行。”

“哦，不过如果独立开发的能力很强，会不会不合群呢，让人觉得没有团队意识，影响整个团队项目的进度和性能呢？”

“不光是独立开发能力强的人不合群，其他能力不行的人有时也是没有团队意识的。我们公司以前有一个员工，并不是很厉害，不过却很喜欢指出别人的错误。我们一起开发一个项目，他对自己的代码不仔细研究，问题多多，却总是喜欢找别人代码上的缺陷，结果惹得团队里都没好气，使得项目的开发过程也不是很愉快，差点耽误了开发进度呢。”

“那如果一个开发人员的技术过于厉害，会不会对整个项目的开发造成些坏的影响啊？”

“不会啊，技术好了怎么还会变成缺点了呢，技术好对整个项目的把握越好，越会朝着团队的目标前进，也许会有些不合群，没有团队意识吧。如果这样的话就比较危险了。”

“对，团队意识说小是小，说大是大！”

“恩，说得很有道理。能力提高下工夫就行，做个对团队有利的人却不容易啊。”

像是在故事里说的，能力再强也要有团队意识，很多高手就是团队意识较差，使得自己瑜不掩瑕，反倒竞争力变弱。高手的无团队意识主要表现在如下两个方面：

- “诸葛亮”型的开发高手，作为 Team Leader 或项目负责人，高手事必躬亲，总觉得别人做的自己不放心，大大小小的事情都要操持。长期下来，自己每次做项目都干了几乎全部的工作，而手下几乎没得到什么成长。小项目还顶得住，某一天接到一个大项目，高手也就慌了，因为再牛的高手也需要时间，而手下人却都眼巴巴地望着他，心想：这下你一个人做不来了吧。
- 一个高手在团队里做队员，如果他是个安分守己，按时完成任务的低调高手，这样还算好，如果高手是个不甘寂寞的人，表现欲比较强，总是不当地批评别人开发中的错误，没人愿意被自己同级的人骂，高手如果总是这样压制别人，肯定影响到队伍的团结，这样整个团队的战斗力就被大大减弱了。

在欧美的团队模式里，比如在前面提到的微软的团队模式中，分工是比较明确的。越是大的项目，越需要这样，这样各个部分可以各司其职，减少其耦合度，比如结合图 1-2，开发人员只需要和产品经理和测试人员互相沟通即可。

日韩企业的团队中，项目分工也是很好，只是过于明细，尤其是外包业务，所以对于开发人员来说只需要做做“体力劳动”即可，而且这样的话团队之间的联系交流程度相对比较小，这种团队不能说是具有独立开发的能力，因为每个团队成员对整个项目并没有很好的把握。

1.3.3 不要停止思考

“蔡佳娃，我们继续刚才的话题，我想要说的是，不管你以后到了公司做什么，都必须记住，不要停止思考，这才是开发人员的 Golden Rule。”

“不要停止思考？”

“是啊，作为一名开发人员，你得明白自己是个脑力劳动者，如果有一天你发现自己的身体比心智还要疲惫，你要小心，是否自己已经变为一个披着脑力劳动者外衣的体力劳动者了。”

“啊，这都有可能啊？”

“是的，所以不管是在团队还是自己独立开发，照搬自己或前人既有的东西可以，但是不要因此麻痹了自己的神经，遇到什么事情总想着从哪搬来凑数。要让自己的脑子时刻处于思想新东西的活跃状态，你才能感受到创新的快乐。”



提示 做脑力劳动时间长了，形成一套固有的模式，不再善于思考是很多开发人员犯过的错误。这对职业生涯的继续良好发展有很坏的影响，读者要多多注意。

1.4 这条路大家都是怎么走的

从编程菜鸟变成业界大牛，这或许是每个走出校门或者踏入职场的人最根本切实的希望。不过对于很多新手来说，这条荣耀之路有时只是一个概念，并没有明确清晰的计划或蓝图。本节将帮助读者把这条路看清。

1.4.1 职位和待遇是怎么升的

“师兄啊，其实我最想咨询你的事情是这样的，一般来讲，一个菜鸟进入 IT 这个行业，如何一步步走上去，把自己搞得很成功啊？”

“这种成功之路一般来讲每个人都不完全一样，大都是能力和机遇的双重作用吧。机遇这个因素很重要，很多 IT 风云人物都是抓住了 IT 这个神奇行业所体现出来的机遇，才一夜之间或者短期之内将自己从默默无闻变成鹊起成名。”

“恩，机遇这个东西很难遇到啊。”

“不是很难遇到，机遇不是公交车可以等到，机遇都是嗅出来的。我们的世界不是缺少机遇，而是缺少发现机遇的眼睛。”



提示 如果在这里再次重复牛顿和苹果的事情，也许会对这本书的创新指数大打折扣。但不可否认的是，牛顿之所以是牛顿，就是他能从苹果看到重力，就像 IT 巨人们能从集成电路中看到未来一样。机遇是给有准备的人的，这句话永远也不会过时。

“师兄你说得很对啊，那我还是先把自己的能力好好搞上去吧。”

“正解，机遇我们有时不好把握，但能力却是实实在在的东西，可以提升。一般情况下，进入公司，升职还是要靠自己的能力。”

“那具体这个过程是怎么样的呢？师兄你描述一下吧。”

“首先你进入一家公司，肯定要从底层干起，先做开发人员，然后崭露头角之后可以做到 Team Leader，负责整个小团队，然后慢慢发展，会给你负责整个项目的开发，也就是类似项目经理，但是只是技术上的负责和领导。”

“哦，那然后呢？”

“然后就是你要考虑的问题了，你是更喜欢做技术还是更喜欢做管理，喜欢技术就是负责整个公司的技术发展，比如技术主管；喜欢管理就逐步走向项目经理或市场等领域。当然做技术主管必须也会管理，项目经理也必须懂技术，只是侧重点有所不同罢了。”

“哦，我稍微明朗一些了。”

“其实根本不用把这个研究这么透的，这些东西你在步入职场慢慢提升自己的时候都会逐渐了解的。”

“哎，我就是想提前感受感受呗，幻想一下自己的美好未来激励一下自己，呵呵。”

其实，蔡佳娃的这个想法非常正常，一位 IT 成功人士回忆起自己的创业经历时曾经说过，当我和几位创业伙伴在北京的地下室里艰难地熬过三年中的每一个日日夜夜的时候，让我们坚持下去的，不是什么宏伟的念头，而是简单地幻想着成功之后的甜蜜和喜悦。

所以，白日做梦不是什么坏习惯，关键是做完梦之后，是苦笑一下了之，还是努力试着让它实现。

“你这个想法倒是不错，IT 行业有时的工作是很让人泄气的，我们有时也这么想，不管会不会变成现实，至少帮助自己把当下熬了过去。”

“恩，师兄说的是。那薪水的巅峰之路是怎样的啊？”

“这个当然和你的职位息息相关了啊。再说不同的公司规模不一样，同样的职位也许会有不同的工资呢。研究这个没有多大用处。”

“哦，师兄那你讲讲你的成就之路吧。”

“嘿嘿，谈不上。其实你们以后去找工作，大多也是去中国的公司，或是外企在中国的分公司，

中国公司里面的职位制度不是很明显，更多的是体现在薪水上的。我就把我当时进 IBM 的经历讲给你听听吧。”（见图 1-4）

图 1-4 是 IBM 中国软件开发中心的一个简单的升职流程图。虽是跨国公司，但其实中国的 IT 公司大概也是这个流程。需要先说明一下，故事中的牛开复同学是通过软件外包的形式进入 IBM 工作的，所以图 1-4 指的也是这种形式下的职业生涯发展过程。

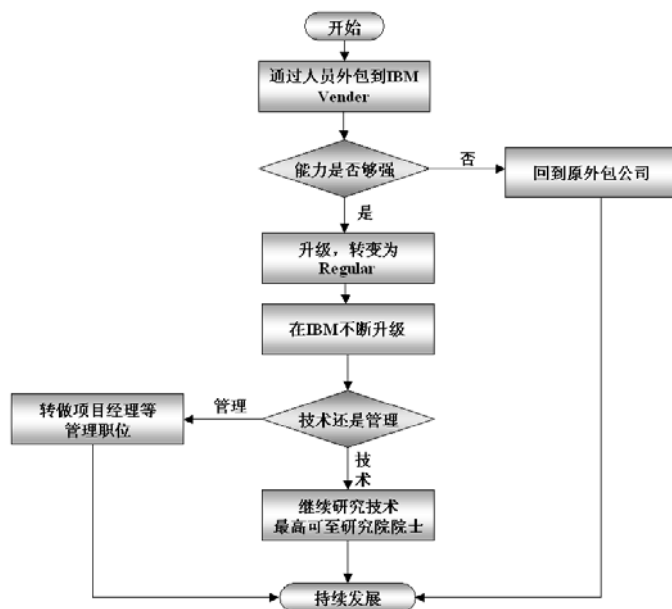


图 1-4 IBM 成功流程图

提示 这里需要提一下的是 IBM 对员工的技术能力是划分等级的，类似于柔道几段那种。分为 1 到 9 级，9 级是最高级别。

1. Vender 阶段

这是进入 IBM 最初的职位，由于是“贩卖”进来的，所以还不能彻底算是 IBM 的员工。但是想要进去也不是那么容易的，一般要求有一到两年的工作经验，能力必须也要符合 IBM 的招聘制度。Vender 一般的等级为 5 级或 6 级，也会有少量 7 级的。

2. Regular 初期阶段

项目开发完成以后，即外包合同已经结束。这时候对于 Vender 一般有两条路可走，能力比较出众的，就有可能被 IBM 聘用为正式员工，即 Regular，这时候就算是一个真正的 IBM 人了；能力并不是很出色的一般继续回到原来的公司工作。

3. 技术巅峰还是管理牛人

变成了 Regular 之后，作为 IBM 的正式员工，就要开始在众多的 IT 精英中求生存和发展了。Regular 继续发展，又会面临两条路可走。

- 第一就是继续做技术，经过努力让自己的技术等级不断提高，IBM 对于高技术人才政策是很多的，会提供最好的环境进行技术研究。当然这里的技术就不再只局限于某个具体的业

务项目了，一般都是一些技术发展方向、新的框架等。最高级别的技术人员是 IBM 研究院院士，这个级别很难达到，因为全亚洲也不过百人。

- 第二是走向管理层，从 Team Leader 到 Project Manager，然后慢慢做到区域总经理，一步一步升上去。最后可能会很少研究技术，主要是研究公司的内部管理、业内合作、投资的运作等关系到公司发展方向的内容。

这些升职经历都只是作为参考，虽然只有短短几百字，但是真正要做到可是需要异于常人的努力的，万丈高楼平地起，做事要有愚公移山的精神。

1.4.2 有干不动的时候吗

青春饭，任何有志于从事 IT 这个行业或者对其有所了解的人都会听到过这种描述，这个原先用在文艺圈的词现在又拿来吓唬 IT 人。

“师兄，有个问题我一直很不清楚，究竟我们这一行能干多久呢？很多人都跟我说这是个吃青春饭的行业。”

“不是你一个人这么问过我了，真是好事不出门，谣言传千里啊。第一个这么说的人肯定是个被 IT 行业淘汰的人。”

“是吧，我也觉得要真是这样，这还不如演艺圈呢。演艺圈都讲究实力派，老练成熟的演员有时更受欢迎呢。”

“呵呵，说的很对，所以你以后要批评所有再提这个论调的人。”

“他们是不是干不下去了所以才这么说啊？”

“并不是只有行内人才这么说的。有这种观点的人或者是个外行人，道听途说，就像你这样；或者是行内不成器的人，自己干不下去了，只好搬出这个说辞做借口。”

对于认为 IT 行业是“青春饭”的行内人来说，干不下去了一般有以下几个原因：

- 对工作无热情，进入 IT 行业只是听说能赚钱，对工资的热情远远高过工作。这样的人很快就无法忍受 IT 行业中踏实严谨的作风，没有工作热情，自然无法在竞争激烈的 IT 行业生存下去。
- 每天的工作是重复性质的，这可能是所在工作环境的缘故。如此长期下去，积极性也会慢慢消磨殆尽；又或者是个人能力达不到，职位也无法得到升迁，自然慢慢懈怠下来。
- 健康问题，这种情况比较少见，不过这种情况也比较情有可原。IT 行业毕竟是比较累的，如果就职的公司喜欢让员工加班，随着年龄的增长，自然健康状况会变糟，最后只好提前离开这个行业。

“蔡佳娃，下面给你讲讲计算机程序之母老而弥坚，终生奋斗在 IT 行业最前沿的故事吧！”

“好，我洗耳恭听！”

“生于 1906 年的美国海军女将军格雷丝·默里·赫珀或许是判决‘青春饭’错误的最好证据。格雷丝·默里·赫珀首先是著名的数学博士，被人称作是“计算机程序员之母”、“编译器之母”、“COBOL 之母”，她是计算机创造初期第一批的程序员。”

“二战时期她在美国海军服役，负责计算机程序方面的研究，期间她为计算机程序设计的发展作出了非常大的贡献，并荣获海军将军的荣誉。退役以后它还在为海军工作，因为，海军竟然无法找到人来接替她的工作。那时，她已经是个六十多岁的老人了。一直到她去世，她都一直在做

着我们这行人认为‘青春’的东西。”

“啊，这真是我们应该学习的楷模！谢谢师兄，我会朝这个方向努力的。”

或许我国的 IT 发展年限还较短，很少会有上年纪的开发人员。美国等其他国家还是有很多研究学术或者研究技术的老当益壮的大牛的。可能很少人会做到像他们那样的声誉和地位，但是至少应该让自己陪着这个行业走到最后，实现自己的人生价值。

对于如何让自己干下去，其实并不难。

- 保持始终饱满的工作热情，很多 IT 巨人的工作热情都是让我们难以理解的，像比尔·盖茨之所以能够建造一座宏伟的微软大厦，这和他对计算机技术的热爱是分不开的。
- 通过做一些不可避免的重复性工作，通过量变达到质变，提升自己的能力素质。坚决不要做那些披着脑力劳动者外衣的体力劳动者，那样很快就会被淘汰。

希望看到这里本书已经说服了各位读者彻底与“青春饭”论调划清界限。随着时间和职位的变化，体力劳动在减少，脑力劳动、知识的积累和丰富的经验却在不断地增加。这些都是时间给予每个人无价的财富。

1.4.3 走的人多了，还会有路吗

鲁迅先生在《故乡》中曾经说过：“这正如地上的路；其实地上本没有路，走的人多了，也便成了路。” IT 行业从一开始的没有路，飞速发展成现在的高速公路。而 IT 界对应文豪的话，却有了另一番解释：“地上本来有路，走的人多了，也就看不见路了。”

“师兄，你说现在有想法从事这个行业的人这么多，会不会路越走越窄啊？我听说现在这个行业的竞争非常激烈啊。”

“那你说高考的千军万马过独木桥这路窄不窄？你不也过来了么？”

“说的也是啊，可是……”

“首先批评你一下，就像上次跟你说那个‘青春饭’的问题一样，很多事情都不要道听途说。不过这次与‘青春饭’有所不同，因为竞争激烈倒是真的。”

“的确，不过干什么应该都不容易吧，师兄？”

“那是，想想看当初 IT 前辈们创业的时候竞争倒是不激烈，可是他们也没什么可利用的资源，没有前人的经验作指导，没有先进的仪器搞发明，但是他们却在艰苦中彰显卓越，为我们开辟了一个 IT 时代；而现在我们虽然竞争激烈，但是却有很多资源可以利用，有很多书籍作为指导。”

“是啊，比起创造 IT 时代的前人们，我们的确是没什么可抱怨的。”

“正解，所以无法在竞争中生存，只是自己不够厉害，失败应该从自己找原因。”

“师兄你说得对，面对竞争我们必须有勇气去赢。”

“真正的高手越是在艰苦不利的环境中，越能表现出超高的战斗力和超强的抗打击能力。”

鲁迅先生还说过：真正的勇士，敢于直面惨淡的人生。目前竞争激烈的 IT 行业还谈不上“惨淡”，所以面对着竞争，任何职场新人都应该拿出本该燃烧的激情和初生牛犊的胆量，在 IT 的竞技场里面都能做个勇士。

不管怎么说，IT 都是一种职业，既然作为谋生手段就会有改行的可能。除非有高到青天的智商和厚如大地热情（如果真是这样的话也早已经是蜚声内外的大学者了），总有感到疲倦或厌倦的时候。升职、转行、创业都可能是 IT 职业生涯的最终出路。

“师兄，继续前天的问题，假如万一我真的在这一行干不下去了，或者厌倦了，我还能有什么出路呢？”

“哎呀，你想的还挺周全。的确，当初和我一起进公司的同事，有些也都已经转行了，留下来的只有一多半。不过人各有志，去留肝胆两昆仑嘛。”

“那他们最后都还混得可以吧？”

“那我就不清楚了，有的上了一层台阶，有的跌了一阶。但都算转行了，对吧。”

在 IT 行业中转行不干的原因很多，最常见的就是觉得无法再有更高的成就（当然不是已经达到顶峰）了，或者总是混不出头，再或者就是要自己单干当老板等。铁打的 IT 职场，流水的人才，这也算是一种良性循环吧。

总结起来，在 IT 行业做到中途离席转行的，一般出路如下：

- 进入高校做老师，选择这一条出路大概是受够了原公司不间断的熬夜加班，不想让自己趁年轻拿命换钱，到老了拿钱换命。不过这条出路也是比较困难的一种，因为现在高校老师这个职位的安逸是很多人都想追求的境界，竞争绝对不亚于 IT 行业。所以想要达到这种要求，硕士、博士文凭不是万能的，但是没有也是万万不能的。
- 努力爬到管理层，这条路比较正统，因为到了管理层，至少冥思苦想的繁重脑力劳动会大大减少，严格意义上讲应该不算转行。但是的确会将开发人员从那些压力中解脱出来，当然这条出路的难易程度因个人的能力而异。
- 转而做培训，IT 行业火爆，竞争激烈，于是这种提升竞争力的培训项目也就有了市场。开发人员转去做培训应该是最简单的一种方式。不过这种出路表面上和当高校教师类似，实则不能保证稳定性，但是退而求其次，况且培训也是待遇不错的选择。
- 自己单干，这种出路应该是最让人向往的，也是最有挑战性的。自己出来单干的原因应该很多，或者是嫌自己挣得钱配不上自己的实力，或者是与公司高层意见相左，或者是想自由自在，或者是有自己的新 idea 等。自己出来单干成功的例子有很多，但是还有更多我们不知道的失败经历。不管怎么说，自己开公司当老板都是很有魄力的。

世界上第一张软盘的设计者艾伦·舒加特就是一个非常好的“单飞”榜样。艾伦·舒加特在 IBM 服务了 18 年，最后毅然地选择离开。舒加特先是在一家公司做高级工程师，随后于 1973 年创办了舒加特联合公司，专门生产软盘驱动器，但一年后他被自己公司的风险投资家解雇了。

1976 年，蛰伏三年的舒加特王者归来，先是宣布研制成功 5.25 寸软盘，之后与人合伙成立了舒加特技术公司，舒加特在这次创业中大展身手，他发明的微机系统接口（SCSI）和研制成功的能与软盘兼容的硬盘驱动器，都成为后来行内的标准。

1985 年，舒加特将公司改名为希捷，谁料之后公司便陷入了困境，希捷公司的客户（包括最大的客户 IBM）都与希捷公司中断了业务，之后舒加特的合伙人也与其分道扬镳，成立了另外一家公司，与舒加特进行了长达十年的殊死竞争。

1996 年，舒加特收购了最大的竞争对手，即自己曾经的合伙人创办的公司，至此宣告希捷成为希捷成为世界硬盘厂商的巨人。不过舒加特注定不能安逸地享受既有的成功，1998 年 68 岁的他又被自己亲手创建的希捷公司强行辞退。被解雇后，舒加特和别人又创办了一家风险投资公司，专门扶持和培养小企业。

舒加特的创业史可谓跌宕起伏。在他几十年的创业生涯中，在遭受重重挫折的时候，舒加特仍然像个勇士般地战斗，不仅在领导公司和创造财富上都取得了重大的成功，还因为自己的天才创造赢得了行内人的尊敬。

- 转而做生意，或者从事其他毫不相干的行业。这种出路就很难说了，因为彻底和自己之前从事的活动无关了，也就无从评论了。

1.5 大公司，小公司

归根结底，所有求职者最后还是会进入公司上班，公司按照前面小节里讲的分欧美、日韩的、中国的、传统的、外包的等。但是最普遍的一种分法就是大公司和公司，简单来说食量不一样，吃的多少和吃的方法也就不尽相同。

1.5.1 大公司爱专才

大公司的概念应该是员工达到 500 人以上规模的公司。大公司里面的团队模式比较正规，因为要管理好数百的员工，除了思想教育，就是纪律了。

“蔡佳娃，还有一个问题得跟你说说，到你找工作的时候，不仅要看公司的国籍、性质，也要看它们的规模大小。”

“师兄的意思是公司给的钱多，小公司给的钱少？”

“不是啦，这都不一定，小公司由于成长较快，需求人才度高，也许会提供比大公司更高的待遇，而大公司都是见过世面的人了，或许你的才能到那里并不会得到重视的。最后再声明一遍，直接决定待遇的，永远是自己各方面的能力。”

“恩，师兄我下次再也不提钱了。”

“好，我们继续说，大公司的机构肯定是比较庞大的，官僚气息也会比较严重，我指的不仅是中国的大公司，国外的大公司也是有官僚主义现象的。”

“恩，这个应该不怕，毕竟长这么大，官什么样咱也都见识过。”

“呵呵，下面讲讲大公司的开发团队，这是和你关系比较大的。大公司因为人多，所以管理一个项目也比较层次分明，每个员工的任务也比较明确。”

“恩，这应该就是大公司的好处吧。”

“不错，大公司对于员工的要求有一点就是会的技术可以不多，但是要专而深。大公司人才很多，你会多了却没一样技压群雄的也没用，我们公司的团队里面就是，小组里每个人都精通不同的技术，加到一起就很厉害，因为这样我们的技术领域既没有盲点，又没有重合。”

“恩，的确是这样，一招鲜，吃遍天。”

大公司的项目管理结构大致如图 1-5 所示。

一般情况下大公司接手的项目规模也必须要大，因为巨人肯定不能靠只麻雀果腹。既然项目规模大，分解开来，每一部分仍然很复杂，所以还会逐步细分。项目分到最后，每个部分所涉及的知识面已经非常狭窄了，所以只需要在这方面精通的人来开发即可。

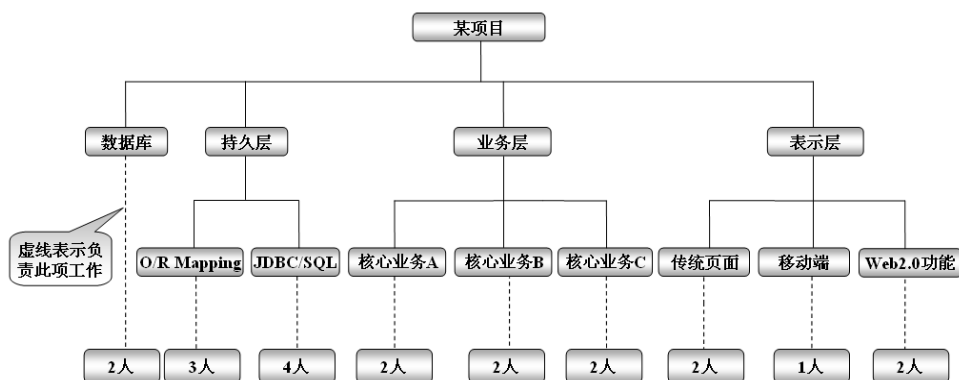


图 1-5 大公司项目管理结构示意图

大公司肯定不会缺少技术人员，但是肯定缺少专精于某项技术的人才。所以只是技术很全面的人去大公司工作会很难过，虽然很多技术都懂，但是总有人比自己更精通。时间长了，自己什么也学不到，反而让上级认为地主家也没有余粮养闲人。

1.5.2 小公司爱多面手

小公司的规模一般比较小，员工在 50~100 人左右。小公司一般没有能力为每种技术都配备一个领域高手，所以小公司的职员虽不能“万事通”，也差不多算是“百事可”了。

“师兄，那你说说去小公司是怎么个状况呢？”

“小公司肯定不会那么人才济济啦，所以小公司更喜欢那些熟悉很多技术的多面手，用大而全的知识来满足客户的需求。”

“看来大小公司的需求还真是不一样呀。”

“小公司的项目管理模式一般比较随意，因为人也不多，有时候整个公司就只有一个开发团队，所以很多事情大家都是身兼多职。”

“这种方式其实也挺锻炼人的。”

“是啊，这种方式让每个人对整个项目都有一个很好的把握，做起来更容易，和团队其他人的交流也更容易。”

“呵呵，师兄，看来大公司和小公司各有各的不同啊，孰好孰坏，不能一概而论哪。”

小公司的项目管理结构大致如图 1-6 所示。

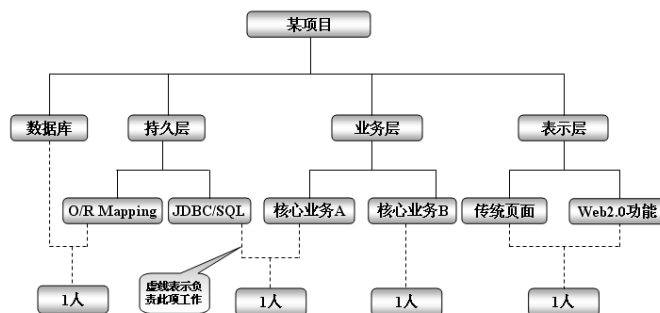


图 1-6 小公司项目管理结构示意图

虽然小公司接手的项目不算大，但是麻雀虽小，五脏俱全。就像演员们常说的：没有小角色，只有小演员。小公司的项目由于规模、资金、开发周期的限制，不可能寻找专精的开发人才来进行开发，而进行项目开发的人员就必须身兼数职，多劳多得一些了。

其实，存在于大公司和小公司之间的还有大量的中型公司，这些公司既追求像大公司那样的专才，也需要像小公司那样的多面手。

正像故事里说的那样，公司规模大小并不是判断自己待遇高低的标准。进了大公司可能道路好走些，但是也很难在那里做到鹤立鸡群，而且门槛也会很高。而小公司门槛低些，上升空间可能会更快，但是技术深度比不上大公司，公司理念和文化可能也不会有很深的底蕴。

1.6 本章小结

本章向读者揭秘了现今 IT 行业的种种“内幕”，旨在让广大读者在读过本书后对风光无限、风头正劲的 IT 行业能有一个全面的了解。当然了，知己知彼才能百战不殆，了解行业更应该了解自己。让自己变得有用和不可替代才是王道，因为哪家的地主都不会有余粮的。



Java 程序员 职场全攻略

——从小工到专家

吴亚峰 著

懵懂菜鸟如何才能成长为一名成功的开发人员？

如何在“血雨腥风”的求职大战中脱颖而出？

职场中只要技术过硬就可以吃遍天下吗？


是学招式重要，还是练内功重要？

掌握了哪些技能，才算是真正的高手？

.....

资深技术专家，娓娓道来。



 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

第 2 章 雾里看花——职场误区

通过前面一章的行业揭秘，读者应该对 IT 这个行业有了一定的了解。但就像海洋拥有宝藏的同时也拥有漩涡暗礁一样，IT 行业也存在着一些误区。“触礁是小，沉船是大”，因此在步入 IT 这个充满成功与激情的世界时，看穿职场的种种误区就显得很有必要了。

2.1 到底差不差钱

都说谈钱会很俗，这应该是本书至少第二次提到钱了。说实话谈钱其实并不一定俗，在经济环境下让自己的财富增加是件无可非议的事情。不过在 IT 这个行业中，尤其是新入行的人，如果把钱看得太重，绝对不是一件好事。

误区：眼中的 IT 更多的是个淘金地，而非一个刺激新奇的技术天堂。处在这种误区的人，并不一定技术很差，而是薪水在其心中的位置远远超过了对技术的渴望。

分析：或许因为是自己这么多年来一直在学校接受教育，总是伸手要钱，让自己不自觉地背上了沉重的心理债务。又或许是因为接触的社会让自己变得急功近利，于是毕业之后迫切需要寻找到一个高薪的工作。

蔡佳娃就是处在这种误区里的人。因为来自农村，可能对钱的感悟更深刻一些，或许是想对辛辛苦苦供自己读书的父母有个实实在在的交待，所以在咨询牛开复的时候，总是时不时就扯到钱上来。

“师兄，我这样是不是不太好啊？我想到一个职位的时候，第一个念头总是挣多少钱。”

“有你这种想法的人还真不少，不过你算是少数的几个能自己看出问题来的。”

“那师兄你说我该怎么去掉这个念头啊？”

“其实想到钱并不是什么坏事，关键是不要把 IT 这个行业仅仅作为一个挣钱的途径而加入进来。把钱看得太重，就把自己看轻了。”

2.1.1 大家都是个什么身价

根治这种症状的第一步，就是让其彻底看清大家到底都是什么样的身价。不过这服药可不好熬出来，薪水这个问题，从哪个角度讲可能都不一样。横向可以分为不同学历，不同技术、不同公司、不同职位、不同资历等，纵向又可以分为不同的时期，这着实是个统计学的大难题。

所幸，这一步并不是最重要的，可以本着保证治愈的原则在统计细节上稍微放宽一些。

1. 起步价

每一个步入职场的新人都必须从试用期这个起点出发，但是就像出租车也会因为城市、车的档次、油价行情等因素定制不同的起步价一样，IT 这个行业的起步价也千差万别。这里统一进行一个假设，即所有人都是能力不低，至少是以优异成绩毕业的职场新人。

“蔡佳娃，为了让你不再陷入钱的误区，我先给你讲讲现在 IT 人的薪水情况吧。好让你彻底搞清楚这里到底能淘到多少金。”

“恩，还是师兄你最了解我。”

“首先谈谈大家的起步价，即试用期的薪水。由于学历、公司、地域的不同，起步价还是有很大差别的。为了方便，假设大家都是顺利毕业，成绩优秀，能力比较强吧。”

“啊，这么高的标准啊！”

“不然怎么样？假设大家都是不学无术、混吃等死的乌合之众吗？IT 行业永远追求的是最高的标准，IT 人也不能这么不思进取呀。”

“是是，那师兄你说吧？”

“首先是学历的不同，硕士研究生一般比本科生的入行薪水高 40% 或者更多。有些公司的某些职位是只招收硕士以上研究生的。”

学历对于一些 IT 方向的在校大学生来说是个比较纠结的话题，往上考吧，怕进去学几年出来跟不上形势掉了队；直接顶着学士学位冲吧，又怕人家以文凭取人把自己埋没了。其实很多时候 IT 公司招聘最看重的还是个人能力，所以不管是什么学历，能力才是最先要保证的。

“啊，看来多读几年书真不是白读的啊。”

“除了学历，大公司和小公司的试用期也是不一样的，大公司门槛理所应当高，所以对于招进来新员工的能力还是比较有信心的，所以放心大胆地给钱。而小公司就不一样了，它们一般不会冒这个险，所以试用期工资就较低一些。”

“是啊，毕竟是大公司，做事肯定要有派嘛。”

“也不能太绝对，一般小公司并没有太雄厚的资金，某一职位的薪水可能要比大公司的同样职位少很多。不过，有些小公司规模不大，但是正处在快速上升的阶段，十分需求人才，所以薪水不一定会比大公司少很多，甚至会更高。”

“是呀，所有大公司不都是从小公司成长起来的嘛。”

“正解，所以小公司也是不能全部小觑的。”

需要说明的是，虽然能力有高低，进入 IT 职场的初期，基本上大家的试用期工资都差不多。这是 IT 行业收入两极分化最不严重的时期，正所谓站在同一个起跑线上。当然还要考虑公司所在城市的差异，因为不同城市的物价指数和消费水平是迥然不同的。

参考价位：2000 元~4000 元。其实不应该设置上限，因为 IT 本来就不是那种中规中矩的行业，一切奇迹都有可能发生，就看天才怎么出招了。

2. 上升价

上升价是指在新员工逐渐熟悉了工作环境后，随着工作时间的增加以及经验和能力的提升，在几年之内给自己带来的薪水上升空间。如果起步价是公司给定的话，上升价就完全靠自己的努力了。这里又该提一提 IT 行业的“一切皆有可能”了，因为真正的差距将在这里拉开。

“师兄，新员工的薪水我大概有些了解了，那么新员工的薪水能涨到多少呢？”

“一般员工在工作了一到两年之后，薪水会提高 6 成或者更多。当然这只是一个平均概念。需

要考虑的是所从事的技术是否具有很高的应用价值以及所在的城市，另外大公司和小公司的增长幅度也是不一完全样的。”

“哦，那是怎么个不一样法啊？”

“大公司的薪水是普遍较高，而且薪水制度和福利制度比较完善，比如 IBM 的新员工本科毕业试用期后每月大概基本工资 5000+，但是完成的任务越多，加成也越大。公司还会为员工设立补助账户，租房补贴和年假、探亲假等制度也都很健全。而小公司薪水可能会相对低一些，但最主要的还是福利、休假制度不很健全，这些其实也是待遇的一部分啊！”

“是呀，有时候工资是不能完全代表待遇的。”

“当然了，上升期能上升多少，最重要的还是看自己的个人能力。”

“说的也是，这个时候不管是人是妖，都是原形毕露的时候了。”

“没错，上升期的身价能增加多少就全靠你自己的能力了。我认识的一个师兄就挺牛，工作了两年多一点，年薪就达到了十八万，而且在房价猛于虎的广州买了新房。”

“哇塞，那他肯定非常厉害吧，技术大牛？”

“也不全是，他大学的时候学习可不怎么样，差点就毕不了业。”

“那他技术方面肯定很拉风吧？”

“怎么说呢。他这个人对于核心 Java 的领悟绝对不一般，不过 Java EE 他并没有深入学习，只是在毕业前研究了一下 SSH 框架和当时并不流行的 Java FX 技术。之后就大大方方地到广州找到了一份工作，五个月后，就跟我们说已经做到 Team Leader 了。”

“的确是不一般啊，看来薪水的提升幅度不仅和个人能力有关，恰当地把握新技术的趋势对于获得高薪也很重要啊。”

上升价时期可是决定一个人能否在 IT 职场淘金成功的重要阶段，上升期始于起步价之后，终于稳定价之前。上升价时期一般 IT 员工都会随着技术能力的熟练和经验的增加将自己的薪水提高到一个不错的水平。

值得注意的是女性 IT 员工的身价上升速度一般要快于男性，或许并不应该感到奇怪，因为世界上最初的程序员都是女的，只是后来计算机程序变得越来越流行和商业化，男性程序员才慢慢地加入到这个行业中来，并且毫不留情地占领了这块多金之地。

IT 世界中存在着悬殊的差距，上升价的高低因个人能力而定，一般是在最初工资的基础上翻 1 到 N 倍。如果碰到的是个天才和一个庸才，那这两个人的上升空间可能会相差十倍都不止，这就是 IT 世界，一个充满差距却又无比公平的世界。

参考价位：4500~不限。

3. 稳定价

稳定价比较简单，就是以个人能力达到事业巅峰之后平缓发展时期的身价，或是已经工作十年以上时的身价。稳定价再往下发展就是自己拉大旗干，不过这种方式的身价不好衡量，所以不做考量。

“师兄啊，那我的工资上升到什么时候算是个头啊？”

“总有一个临界点，过了那个点，你的工资就不会再有大的涨幅或跌幅了。”

“那是个什么样的临界点啊？”

“比较平淡的情况下，就是工作十年以后。除了极少的大器晚成之外，十年的工作会让你的事业趋于平稳。因为如果你在这个行业中十年都没办法让自己的工资达到一个非常高的层次，你基本上在今后的十年里也不会有比较大的动静了。”

“有道理啊，张爱玲不也说过嘛，出名要趁早。”

“呵呵，差不多。稳定价应该算是薪水飞涨的最后一站了。这个时期薪水一般都是论年薪了，而且一提年薪一般都是十万以上了。”

“恩，我希望我的稳定期来得晚一些，好让我的薪水多上升一些。”

“傻啊，上升价可不是靠上升期的长短决定的。好比一条直线，决定它上面点的 y 值可不只有 x 值，斜率也很重要啊。”

稳定价的时候已经到了个人事业发展的平缓阶段了，一般职位都在项目主管、经理之上了，或者已经成为某一个技术领域的专家了，而这个时期的薪水一般只和专业领域及职位有关了。

这里需要指出的是每个人的人生追求都不一样，对成功的定义也就不同。因此稳定期的定义也出现了分歧，大致分为以下三种：

- 像李开复、唐骏那样的 IT 高级管理人才，他们缔造了一个又一个的 IT 帝国，谁也不知道他们心中的火什么时候才会熄灭。总之他们肯定会锐意进取，不断创造着 IT 的神话。
- 有些人不擅长也不喜欢管理别人，就喜欢管好自己、研究技术，这些人最后会成为公司的技术总监、首席科学家或者行业内的专家。
- 深知“一分耕耘，一分收获”的道理，知道人家的高薪是忙碌打拼出来的；本着“知足常乐”的原则，自己拿个 5000~8000 的工资也就满足了，这并不是不求进取。其实知道如何收手，有时也是很难做到的。



提示 就像本节中出现最多的一个词“一般”一样，每个人都不是别人的复制品，每个人都有自己的独特地方。对于富有冒险精神的 IT 人来说，本节的薪水介绍仅仅能作为一个参考。自己的价值，最终还是需要自己来创造。

2.1.2 给自己估个好价

看了大家的身价，想必读者心里也开始盘算自己的身价应该定位为多少了。准确地给自己一个定位，不仅会让自己找到合适的职位，更能让自己以最好的姿势起跑。当然，不要太早给自己确定最终的定位，现在的你还有很多准备升值的空间与时间。

“师兄，看了各个时期 IT 人的身价，我发现有必要为自己也定一个合理的价钱了，呵呵。”

“是呀，知己知彼，方能百战不殆嘛。了解自己要像了解敌人那么透彻才行。”

“那应该如何为自己的能力打分呢？”

“很多情况下，我们并不是放在超市里贴个标签写着价格的商品，而更像是菜市场里被人挑来挑去还不时威胁‘一块二卖不卖，不卖我走了’的南瓜或是土豆。我们中很少有人能做到让人蜂拥抢购的程度，能做到展览拍卖的就更加寥寥无几了。”

“说的也对，现在的 IT 行业竞争实在是大啊。”

“所以这个时候对自己的评估并不是我‘会’做什么，而是我‘能够’做什么。两者是不同的，‘会’做什么是对自己现在能力的定位，而‘能够’做什么是对自己将来能力的肯定。要对自己未来的能力有所预见。”

“那到底估价是为了什么啊？”

“研究了这么多关于职场的问题，想必你能为自己的未来勾画出一份蓝图吧。估价，从某种意义上，来说更是对自己的一种规划。你为自己估的什么价位，你自然会让自己时时刻刻的表现都符合自己对未来的规划。”

“恩，说的也是啊。”

上述都是对还没有步入真实职场的人而言的，因为他们还有为升值而进行准备的充分时间。一旦真正到了求职的时候，就只能是结合当前的行业形势，结合自己所学专业在市场上的流行程度，结合自己的能力水平来给自己估价了。那时，如果之前的准备不充分，职场不认可就只能一跌再跌了。



提示 读者在给自己估价的同时最好把准备工作做充分，这样才能让自己的估价得到职场的认可，才能让自己的估价成为现实。

其实有的时候给自己估价除了要看自己的能力，还要看从事的技术方向，即选用的编程语言、面向的操作系统或程序接口等。有时候由于方向选择错误，就算自己再有真才实学，也是会感慨自己英雄末路，无力回天。

1. 不同编程语言的收入

本书在 1.1.3 节中向读者介绍了当今 IT 行业编程语言平台的人员分布，下面继续向读者介绍不同编程语言平台的收入分布。

“蔡佳娃，还记得我们之前讨论过的 IT 行业编程语言的流行程度分布吗？”

“恩，记得呢。就是各个编程语言的排行吧。”

“对，现在就根据每个语言方向看看它们的财富含量。”

“不是三百六十行，行行出状元吗？”

“话是这么说，不过我们现在讨论的主要是面向大众的跟薪水有关的行情问题，而你的状元想法或许只对于少数创造时势的英雄管用吧。”

“哦，那师兄你来说说吧。”

根据编程语言平台薪水分布图(图 2-1)可以看出，从事不同编程语言的工资分布有如下特点：

- 一般来说，越流行的语言，其相应的薪水也会越高，这个道理应该很容易理解。一门语言流行起来，研究的人就会增多，比如 Java 和 C/C++等，自然会将这门语言发扬光大，使其越来越完善。这样就扩大了其应用领域，自然收入也会升上来。
- 有些语言虽然流行度不够，但或许是新生语言（如 C#诞生不过六七年的时间），又或许流行度不高（就像图中未列出的 Python、Delphi、Pascal 等），这些语言或者发展很快，或者在某一些方面的应用有其他语言无法比拟的优势，所以薪水仍然很高。
- 有些编程语言虽然流行度很高（如 VB 等），但是由于主要功能或效率的问题，处理的是

小型应用，并没有在一些大型企业级开发中得到青睐，所以薪水就自然无法和其他语言媲美了。

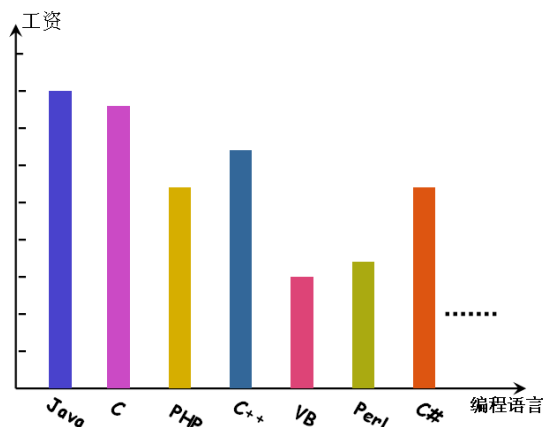


图 2-1 编程语言平台薪水分布图

图中的编程语言收入排行只是相对性的总体比较，因为毕竟鸡窝里也可以飞出金凤凰。同时，这里的薪水情况也只是作为参考，随着时间的推移也可能会发生很大的变化。

2. 面向不同操作系统收入

面向不同的操作系统进行开发，其收入也会有很大的不同。不过一般来讲，选用的编程语言在一定程度上决定了要面向的操作系统平台，比如 C#、VB 目前只能用在 Windows 平台下，而 Java 可以跨操作系统平台进行开发，Perl、PHP 等脚本语言也具有很大的跨平台特性，而 C/C++ 可以面向不同操作系统平台进行开发。

总体来讲，根据中国 IT 行业的现状，面向 Linux 和 Solaris 等系列操作系统的开发要比 Windows 平台的收入高一些。因为前者大部分是用来解决一些高端的大型企业级应用，而 Windows 是因为是从 PC 起家，所以在面向中小型的项目应用时有很大的优势。

有多大本事，就有多大口气。老话说得好：“没有精钢钻，就别揽瓷器活”。给自己估价并不仅仅是让自己更好地卖出去，更重要的是能更好地实现自己的价值。

2.1.3 先挣的是资本，后挣的是钱

赌场上有句话：“先赢的是纸，后赢的是钱”，这句话的意思就是赌博要沉得住气，不要在意开始时的输赢，赢在最后才算是真赢。如果把这个道理告诉那些一味追求眼前高薪的 IT 人身上，或许会对其有所帮助。

“师兄，你先是给我说了说 IT 这个行业到底能淘到多少金，又告诉我应该对自己有个精确的定位，并且对未来有个清晰明朗的规划。那么，在选择职业的时候，究竟对于薪水应该如何看待呢？”

“总体的原则是在进入职场初期，以积累成长为主，以追求高薪为辅。IT 是个成长极快的行业，金钱永远只是飞速发展的技术的附属产物，那些追求高薪的人都是本末倒置了。单纯追求眼前高薪，等 IT 行情一变，自己的金山也就沉入海底了。”

“所以，师兄你的意思就是在初次求职的时候不要太注重薪水，要选择那些自己上升空间大一些的公司喽。”

“对啊，要知道先挣的是资本，后挣的才是钱。努力把自己从菜市场的南瓜土豆变成展览会上的拍卖品，才是对自己最大的尊重，也是对钱的尊重。”

“钻石恒久远”这句广告词用来形容 IT 行业实不为过，但是后一句“一颗永流传”就很不恰当了。或许真的存在可以一劳永逸的事情，但应该不会太多地在 IT 这个行业内发生，在 IT 行业注重积累资本永远是正确的做法。

或许那句“授人以鱼不如授人以渔”用在这里是最恰当不过了，有再多的鱼都不如会捕鱼，有钱不一定有再挣钱的能力，有挣钱的能力才会永远有鱼吃。

2.2 谁给我解决户口问题

户口这个问题，或许有些读者并不是很了解，找工作会和户口有关系吗？有的读者或许已有所耳闻，但仍不解户口这个老实巴交的东西为何会在求职路上横刀立马。

误区：对户口问题处理不当，不是因为当地户口怕被拒绝不敢去求职，就是求职时张口闭口要求单位解决其户口问题。认为户口和学历能力一样重要，甚至更重要。

分析：或许是受了前辈人言传身教的影响，或许是经历过户口的难题已经对户口这个问题伤透了心，总之在求职的过程中，遇到户口问题就如临大敌一般。处在这种误区的求职者，大都对于户口问题并不了解，只是道听途说，其实户口这个政策问题国家每年都在不断改进。

2.2.1 讲讲户口的故事

户口，就是用来标识一个人的身份的。户口主要分为集体户口和居住户口，集体户口是指户口关系挂在所在单位，如果离职或者单位倒闭，就不太好处理；而居住户口是根据居住所在地而建立的，只要不主动迁移，居住户口是永久存在的。

户口问题主要就是指获得某一城市的居住户口很困难，按说这个问题不应该牵扯到职场里面。随着人才的频繁流动，尤其是在中国的一些大城市（如北京、上海等地），使得上述地区的居住户口的获得变得困难重重。

“师兄，你现在在北京工作，户口是不是北京的啊？”

“不是啊，怎么啦？”

“那师兄你不赶紧置办置办啊？你现在可是外来户口啊。”

“这个东西需要着急吗？你想得太多了吧？”

“怎么会呢？户口问题不是很重要吗？师兄你现在不是北京户口，当心以后评职称、结婚啥的有麻烦哦。”

“这就是你对户口的想法啊？看来我的师弟对 IT 这个行业的错误观点还真不少啊！”

“啊？这还算错误啊？户口不是一个很让人头痛的问题么？”

“是，也不是。还是听我给你慢慢道来吧。”

“那师兄你赶紧说吧。给我扫扫盲。”

“首先给你讲讲过去的故事吧。几年前，户口或者真的是个大问题，当时因为北京、上海等大城市就业机会多，所以很多高级人才、海归人才、务工人员都涌入了这些繁华的大城市，慢慢地也就带来了管理治安方面的问题。”

“是啊，人多了就是不好管。结果呢？”

“比如你拿了一把糖果，结果一千个人冲向你，你会不会把糖果捂得严严实实呢？结果就是这些大城市把户口政策卡得很严，人们想把户口落在那些大城市就很难很难了。”

户口政策卡得严的另外一个原因和城市的发展规划有一定的关系，曾经有一段时间这些大城市并没有意识到引进人才的重要性，主要还是偏重于引进投资。对那些回国创业的‘海归’和其他满腔热血的高级人才并不支持，有的时候还会频频刁难。

不过最近几年随着社会经济不断发展，各地也越来越重视人才的引进，相应地也推出了很多人性化的户口政策，如上海的居住证制度等，后面的章节会详细介绍，这里读者不必过分担忧。

“师兄，户口这么难落实，岂不是让人伤心哪。”

“是啊，所以那个时候户口是个抢手的东西啊。因为很多福利和社会待遇都是只对本地户口的人才有效的。没有当地户口做很多事情都深受限制，所以很多人因为没有户口而没办法把家人接过来，也有很多人无法让自己的孩子在自己工作的地方上学。”

“这就是职场的户口问题吧？”

“还不算是，由于户口不能忽视的重要性，很多公司在招聘员工的时候也就加上了条件，比如本地户口优先等，这就是户口的问题。想就业，有能力还不够，没户口就得歇菜。”

“是啊，真的是很残酷的一段时期。那后来呢？”

“后来，像北京、上海等这些大城市开始意识到人才对于经济发展的重要性，而且关于户口制度的不公平也怨声载道，便逐渐开始放宽户口政策，你应该不时地听说某某明星通过什么优秀人才入境计划变成香港人的新闻吧？”

“恩，那也算是一种引进人才的策略吧？”

先把一个人的居住户口迁到某一个城市，这样对这个人而言以后不管到别的任何地方都不算是自己的地盘，都是异乡。再加上中国人对家都有比较深的依恋，所以转来转去还是会回到户口所在地的城市。因此这种引进人才的策略，相当实在有用。

现在很多城市在人才缺乏时也会大打户口牌，这在很多时候都能取得不错的效果。当然把前些年很金贵的户口作为招揽人才的橄榄枝，也说明当下社会对人才的重视了，作为 IT 精英的我们也应该倍感欣慰了。

“呵呵，原来是这样啊，那这么说现在户口还是很好办的了？”

“看看，又犯傻。就算政策一直在开放，但永远是处于狼多肉少的局面。所以就目前来看，户口应该还是一个困扰，并且是长期困扰中国职场人的问题。”

“那不就像我一开始说的那样喽，师兄你应该抓紧把户口落实了呀。”

“我还没说完呢。户口是比较难办下来了，但这并不是说我们这些非本地户口的人在这些地方就享受不到同等的待遇了，现在我几乎是感觉不到一点不自在的地方。我说你还是不要研究这些东西了。户口这个问题现在已经退居二线了，你大可以放心地四处奔跑找工作了。”

“呵呵，那敢情好啊。我就怕到时候我这个外地户口被人拒之门外呢。”

就像《天下无贼》里葛优大叔说的那样：“21 世纪什么最重要？人才！”。所以为了让自己更加具有竞争力，北京、上海等大城市都对户籍政策实行了不同程度的开放，就是为了广纳贤才。尤其是对于 IT 这个所有人都想引领风骚的强势产业户口政策更加倾斜，因为整个 IT 产业说到底实实在在的资产就是人才。

随着国民经济发展所带来的频繁交流和人员流动，各地政府正在逐步将原住民和非本地居民进行无差别管理。所以在公司暂时安身工作并不是一件难事，虽然拿不到户口，但是一些类似于暂住证、居住证等过渡临时性质的户口政策还是让类似“北漂”一族的人感受到了本地人的待遇。

不过毕竟只是“暂住”，比起名正言顺的“居住”来说，还是有差异的。拿不到居住户口，在大城市要头疼的问题主要有以下两个方面。

1. 子女教育问题

这个话题对于刚刚步入职场的人来说应该还是比较遥远，纵观各地的“暂住”政策，基本上对于子女的义务教育，即中、小学教育，是给予与本地居民同样待遇的。

而且，从某种角度上来说，非本地户口的工作人员子女在义务教育方面享受到的好处可能比本地学生还要好一些。比如，外来子弟在享受义务教育的时候，是根据居住地就近入学。因此如果想换一个比较好的学校，重新租房搬一搬家，更新暂住证就可以了。实现“孟母三迁”，对于外地子女来说，或许更加容易。而本地人要想这样就只有买房子、迁户口才行了。

需要注意的是，这种待遇一般也只是持续到高考前。当到高考这个人生转折点的时候，就暴露出本地户口和非本地户口的差异了，不过那至少是你从业二十年后的事情了。对于一个 IT 人来说，如果二十年都没有在逐步放宽的户口政策下搞定本地户口，那么这个人也应该好好反思一下了。

2. 特定社会福利问题

社会福利制度也是非本地户口目前还不能完全享受到的待遇。虽然政策已经很开放和统一了，但总是要优先保障一些有居住户口居民的特定权益。拿北京来说，比较典型的社会福利制度就是经济适用房的购买，这个待遇目前还没有对非北京户口的人开放。

不过看看经济适用房的申请资格：家庭年总收入在 6 万元以下，即家庭每月总工资在 5000 以下，对于立志于要在 IT 这个竞技场赚得一席之地各位读者来说，经济适用房的目标，或许是没有太大价值了。试想两个人月收入不足 5000，还是 IT 精英吗。

2.2.2 各地户口政策面面观

户口问题在很多情况下都是集中在发达城市，比如北京、上海、广州等地。而户口政策是指导如何取得当地居住户口的重要指挥棒。

尽管各地的户口政策逐步开放是一个大趋势，然而就目前的情况来看，北京和上海等地的户口仍然是针插不入，水滴不进。这里向读者介绍一下两个既是 IT 中心城市，又是户口难求城市，又是房价高高在上城市的户口政策。

1. 北京

“蔡娃娃，说了这么多，还是给你具体讲讲各个地方的户口政策吧。”

“恩那，师兄，好让我心里有点数。”

“先说说北京吧。北京应该是北方 IT 的最中心城市了，北京的居住户口算是很难办下来的。简单来说，花二三百万买套房，你就可以有机会名正言顺地做一个北京人了。”

“啊？这么遥远的目标啊！”

“北京的户籍管理特点就是拿到居民户口很难，但是其尽量保证拿到的和没拿到的两个层次的人都能享受到基本同样的待遇。再有一个就是对于高科技或短缺人才提供特权，如果你可以让北京和上海为了争你而大打出手，那么你就可以选户口了，呵呵。”

“哈哈，师兄，你太会开玩笑。”

目前北京的居住户口应该是最难搞定的，北京的户口政策按照传统的方式——指标发放，每年会将有限的户口指标发放给企事业单位等机构。能不能获得，就需要看这个城市需不需要你。或者是这个城市需不需要你所在的单位，以及你所在的单位需不需要你。

北京的户口政策还有一些是面向有亲人在北京想要投靠过来的情况，如娶北京 mm 或嫁北京 gg，不过这对投靠人的年龄或婚龄有一定的限制。而对于那些更多的没有投靠对象，打算赤手空拳为自己打一份天下的 IT 新人来说，要拼得一纸户口，短期内的确不太好实现。

困难指数：★★★★★

2. 上海

“说完京城，我们再来谈谈上海，上海的户口刚刚改革过，比较值得欣慰的是上海的户籍政策抛弃了一直以来的指标投放方式，实行条件准入制度。”

“哇，那是什么条件呢？”

“主要一个原则就是持上海的居住证，也就是暂住证在上海居住满 7 年。其他的就稍微有一些次要了，比方说参加社保及医保、缴纳个人所得税、没有受过治安处罚和不曾参加犯罪活动等。”

“俺是个守法的好公民，这些事肯定不会做的，呵呵。”

“还有一个条件就是你的专业技术必须达标，如果没有高级职业资格，至少也应该是中级以上的技术人才，否则你还是不能拥抱上海的户口。这个户籍政策就是想办法吸引高级人才，提高上海在全国乃至国际经济中的竞争力。”

“看来上海对于户口的开放比北京要前卫啊！”

不过这个政策还是有一定限制的。每一年都有一个控制指标，只为符合条件的申请人按顺序办理，超过控制指标后的申请人延期到下一年。如果不加控制，大量上海人‘诞生’，对社会资源会造成很大的压力。

上海在放宽户籍制度、广泛吸纳贤才这一方面做得的确不错。以前户口问题大都和购买商品住房挂钩，而上海最新的户籍政策将两者彻底分离。对于那些已在上海打拼数年却仍感陌生的“沪漂”们，着实是个很大的激励。

困难指数：★★★★☆

2.2.3 别怕，咱有暂住证呢

各个城市的户籍政策在逐步开放，不过变化程度最大的，还是对流动户口的扶持和重视。虽然各地的户口政策不完全一样，但是所有城市大都对暂住户口非常关照。从一开始的管理人口流动，到现在的面向流动人口服务，下面就向读者介绍一下北京、上海这些大城市的暂住政策。

其实，在暂住证和居住户口之间，还有一个证叫做工作居住证，或者叫居住证。暂住证和居住证有本质的区别，前者是由公安机关颁发的，后者是由城市的人事机关颁发的，所以含金量有很大不同。居住证可是相当于“准户口本”，所以也有“绿卡”之称。

但是随着各大城市暂住证制度的优化和改进，暂住证和居住证之间的差异逐渐缩小。目前暂住证只是相对居住证离居住户口要稍微遥远一些，其他方面两者所享受的待遇是基本相同的。随着政策的改变，暂住证和居住证可能会合二为一。

1. 北京

“唉，师兄，上次你跟我说了说北京和上海的户口政策，看起来不管怎么说我们这些菜鸟肯定还是以暂住证开始的吧？那你就给我讲讲暂住证的政策吧。”

“恩，先谈谈北京吧，北京的暂住证制度可谓源远流长，最初的暂住证是分为 A、B、C 三个等级，C 本是最普通；B 本就有些稀有；A 本就更加难求了，基本上 A 本就跟北京人差不多了。”

“哇，那个时候肯定是困难的那段时光吧？”

“是啊，后来这个就慢慢淡化了，现在随着北京市逐步对流动人口和本地户口实行统一对待的制度，很快将不会对两类人进行区别对待了。这样一来，流动人口在北京享受到的待遇就大大提升了。”

北京的居住户口虽然是最难拿下的，但是北京的暂住证政策却是让人欣慰的。暂住在北京的流动人口，除了在前面介绍过的子女高考问题和特定的社会福利问题暂时无法解决外，其他方面均可以放心不管，安心享受在京城的美好生活。

而且，从 2009 年开始，暂住证也不再是强制办理，一切皆自愿。这应该是对在京城默默工作的流动人口的一项最大的安慰。

2. 上海

“师兄，那上海的暂住证有什么惠民政策啊？”

“上海的暂住证制度和北京就很不一样了。上海已经取消了暂住证，将其归到了居住证里面。而居住证就分为人才引进和暂住人口。”

“那这两者有什么区别啊？”

“人才引进的居住证享受的待遇基本上和上海户口的居民一样了，而属于暂住性质的居住证享受的待遇就很有限了，社会保险、子女教育等问题都是很难解决的。”

“看来同样是拿着暂住证，还是在北京好啊。”

“是啊，上海的户籍制度放宽了，居住证制度稍微痛苦一些也还是可以理解的。”

上海居住证制度中的人才引进类应该是属于上海市的“绿卡”了。人才引进的条件并不是特别高，主要是申请人的学历必须为学士以上、所在公司的规模如注册资金在 100 万元以上，个人

身体健康且按章缴纳各种税费等。

IT 属于很多城市包括上海在内的紧缺专业，所以 IT 人肯定要向着人才引进类的居住证前进，这样才能彻底心无旁骛地为这个城市和自己的未来努力了。

2.2.4 户口问题小结

户口其实就是人事关系和居住地的总和，前文中提到的集体户口和居住户口的概念并不一样，集体户口拥有的只是人事关系，所以要想蜕变成居住户口还需要有一个固定的居住地。

户口政策年年在变，也许在可以等到的将来，中国将不存在户籍制度，所有人不管在哪里都不会有异乡的感觉。说到引进人才，IT 是包括北京上海在内的很多大城市比较感兴趣的行业。而高级的 IT 人才就倍受追捧了，所以不管户口问题变成什么样，对于人才永远是欢迎的。

2.3 我们不是爱加班

加班，就像前面章节中提到的，是很多人对于 IT 这个行业的首要印象之一。随着 IT 渐渐深入到每个人的生活，加班也渐渐成为很多 IT 新人对于这个行业的误解。

误区：是个 IT 公司就会加班，不爱加班的开发人员不是合格的高科技人才。IT 开发人员的高薪，都是加班加出来的。

分析：就像所有的误区都是由于不了解造成的一样，IT 可不是个夜战之国，开发人员也不是猫头鹰。弄清楚加班的根本原因，才能让自己不再受加班之苦。

2.3.1 常态加班是为何

加班，即在正常的一天工作结束后，仍然由于自愿或非自愿的原因延长工作时间，在公司或者家里继续工作。现在加班是个再普通不过的现象了，很少有人没有加过班。不过当加班已经成为一种常态，就需要谨慎研究一下了。

“师兄，进入 IT 这个行业，是不是必须得有良好的加班意识啊？”

“良好的团队意识我倒是听过，良好的加班意识是什么？”

“就是首先心理上必须对加班有正确的认识，承认加班的长期存在性，接受加班，乐于加班。”

“啊？我好像从来没听说过唉。”

“其次呢，身体上要对加班有足够的抗打击能力，身体要能承受经常性的加班，还要保证加班时的工作质量。”

“哇，你们这些没下过水的倒是对游泳姿势研究得不少啊，呵呵。”

“不对吗？师兄你加班不多吗？”

“加班可是有很多原因的，你的这种加班意识也太太无畏了。”

“啊？那具体情况是什么样的啊？”

“别急，由我来给你分析分析先。”

下面针对 IT 界的现状，向读者介绍常态加班的种种不同。

1. 自愿型

“常态的加班分为好多种，首先第一种就是自愿型的，就是公司没有要求加班，自己的任务也没有到非要加班不可的地步，只是这些人对于工作实在是太有热情，这种人属于‘衣带渐宽终不悔，为伊消得人憔悴’类型的。”

“啊，这种人加班应该很享受吧？”

“是啊，完全自愿的加班肯定是双方都没有怨言的。公司肯定不会阻挠，而自己又是主动地拼命，当然是一种双赢的策略。”

“不过这样的人应该不多吧？”

“是不算多，很少有人这么用心的，毕竟身体不是铁打的，再忙也要休息的。”

自愿型加班的人，一般做事比较认真投入，并不是非加不可。只是不加觉得可惜，不加就遗憾。就像玩电脑游戏，看看表，不早了，下线吧；然后又对自己说：再玩十分钟，做完这个任务就下线。十分钟后又会再有其他的借口。

自愿加班并不是什么坏习惯，只是项目任务进行到某一点，思路好不容易理得非常清楚，不想设个断点明天接着再来。如果不一鼓作气将项目一举拿下的话，等到“再而衰，三而竭”的时候，还得去走回头路，这可是比较浪费时间和精力。

不过自愿加班并不总是对公司有好处，人都不是金刚不坏之躯，精力是此消彼长的。晚上用了功，白天肯定要略显疲惫，有可能会在公司的正常业务活动中力不从心，影响正常工作。

加班痛苦指数：★☆☆

加班感受：痛并快乐着！

2. 主观被动型

“蔡佳娃，刚刚说的是自愿型加班，这种加班我曾经尝试过，不过只有几次，毕竟要保证第二天的正常上班。”

“是啊，要不然光做好事还不落好。”

“不过接下来的这种加班类型就不怎么美好了，那就是主观被动型加班，我可不提倡你以后这样干工作。”

“那主观被动型加班是什么样的啊？”

“主观被动型加班是公司没有要求加班，但自己也不是自愿加班。只是自己的当天任务还没有完成，所以不得不去加班了。”

“那这种加班就很不舒服了。”

“岂止是不舒服啊，首先是没办法才加班，其次加班也不一定能够出效果。这种人不像刚才说的自愿型的，自愿型的加班至少是有米可炊的巧妇，而主观被动型的加班就是无米可炊的拙妇了。”

“是啊，这种加班实在是不可取啊。”

“被动的加班效率一般都不高，结果又将工作拖到了第二天，第二天晚上自然还得加，然后一直拖，就变成了常态的加班了。”

主观被动型加班很不好，主要还是个人的工作态度和方式不正确。究其原因，有如下两点：

- 脑子懒，不懂得思考，把任务仅当作工作来对待，而不是一项挑战。总是用笨的方法来解决问題，而不去花点时间研究更新更快的方法提高自己的工作效率。
- 不学习新技术，用笨的方法也就算了，还用陈旧的技术。这么一来开发项目肯定如老牛拉破车，效率低下，质量还不高。

加班痛苦指数：★★★★☆

加班感受：痛苦，痛苦，很痛苦啊！



提示 磨刀不误砍柴工，多花些时间思考，多研究些新技术，这不是浪费时间，是在改进自己，让自己远离这种费力不讨好的加班。

3. 客观被动型

“好了，现在来说说最后一种加班，客观被动型。”

“客观被动型是否也属于非自愿的加班，只是外界原因使然的对吧？比如公司强制加班？”

“恩，差不多，这种加班一般是客观原因造成的，不只是公司的强制，有时候公司没有强制加班，但是当加班已经成为一种作风，一种公司文化的时候，就在所难免了。”

“啊？还有把加班作为潜规则的吗？”

“有啊，最典型的就是日系的公司，在那里按时下班是要遭到上下员工的鄙视的。你就是装也要装到大家陆续开始走的时候再离开公司。中国也有很多公司是这样的，别人都加班，就你一个人大大摇大摆回家，你自己也不舒服，只好留下来共患难。”

“哎，当加班成为一种习惯，不加也得加啊。”

“是啊，这种加班也不是非常累人，不过公司强制的加班就比较难受了。有的人完成了自己的任务，但是由于种种原因，必须还要做一些分外的事情。”

“那加班费还是有的吧。”

“钱这个东西就完全看情况啦。有的公司会算加班费，有的则不会；有的公司会体贴地放你第二天半天假，有的则不声不吭，表示明天接着早来。”

客观被动型加班的原因有很多，有一种就像故事里提到的“加班成为一种习惯”，这种加班原因很简单，就是公司的工作氛围使然。其他的客观被动型加班原因大致如下：

- 员工的问题，有些员工工作不够努力，或者是眼高手低，夸下海口而又隐瞒军情，又或者属于主观被动型加班。总之这些员工由于自身的缺陷耽误了整个项目的进度，到最后只好让所有参与项目的其他员工共同买单。
- 公司问题，每个员工都干得很好，倒是公司管理不善，对于项目运作的规划没有做好，只好最后让员工一起受累。
- 公司没有带一个好头，管理失策，偏偏用人不善，员工也不合格。这种问题加问题的状况使得公司只能彻夜灯火通明，员工挑灯夜战了。

加班痛苦指数：★★★★☆

加班感受：虽然并非自愿，但是考虑到大家都在加班，只好随波逐流。由于不是孤军奋战，至少内心要平衡许多。而且加班也不会完全没有补贴，只是要给别人擦屁股，怨言还是有的。

2.3.2 你为什么加班

“说了这么多加班的事，那师兄你是为什么加班哪？”

“我还是比较欣慰的，我们公司很少会加班，就算有也不是主观被动型加班，主管分给我的任务我都完成了，只是偶尔被公司强制加班。”

“那是公司的制度不好吧？”

“倒也不是，只是总有一两个人，能力不是很出色，但总是硬着头皮扛着，主管分配完任务问大家有没有问题，他们也不说。但是不会做是改不了的，只好一边瞒着上级一边自己找方法解决，主管不知道啊，以为他们和其他人一样都在努力工作。”

“但其实他们确在努力工作啊！”

“话是这么说，等到了最后，主管说了，大家把做的都拿过来，我们整体调试一下吧，然后他们的部分没有完成，没办法，大家只好加班让他们的进度追上我们。”

“是啊，那种加班滋味肯定不好受吧。”

“相当不好受啊，可是也没有办法，总要以团队目标为主吧。”

了解完加班的种种原因后，如果你正在饱受着加班之苦，就应该思考自己是为什么加班了，究竟属于哪个类型。不同类型的加班滋味可是大不相同的。

- 自愿型加班，这种加班方式感觉还是蛮爽的。有些类似癫狂状态的梵高之类的艺术家，对技术的热情可以让自己在工作上表现得更加出色，更容易让自己脱颖而出。不过这种狂热的加班最好还是不要太频繁，否则身体还是会顶不住的。
- 主观被动型：这种加班其实就是哑巴吃黄连，而且就算能说也不可以说苦，因为都是自己的懒惰和不合理利用时间造成的。而且这种加班效率不高，有时候操之过急反倒会忙中出错，所以作为开发人员，这种加班现象是要严格杜绝的。
- 客观被动型：这种加班首先是被动的，而且很有可能加班的起因不在自己，自己只是来“救火”，所以怨声可能会大一些。但是大家都在努力，所以相比主观被动型加班，熬起夜来也要稍微心安理得一些。

2.3.3 让自己不再加班

“蔡佳娃，说到最后，你一定应该知道自己以后如果真的走上 IT 职场，该如何让自己避免常态加班了吧？”

“恩，只是知道了不同加班的原因和后果了。那到底如何让自己不再加班呢？”

“其实很简单，说说客观原因的加班，即由于公司或其他员工造成的常态加班。如果你不能选择一个好的工作环境，只好让自己尽量接受这个事实，或者通过自己的努力去试着去改变公司的环境，实在不行就只好走人了。”

“是啊，那主观原因的加班呢？”

“主观原因的加班如果是自己完不成任务，那可真要下大工夫让自己的能力提高起来，否则总是这样熬夜过关，根本无法从本质上解决问题。不仅对身体不好，时间长了可能连熬夜的机会都没有了，IT 行业淘汰一个人可以很快的。”

“的确是啊，逆水行舟，不进则退。”

“对于那些工作太积极的熬夜，我个人还是比较赞同的，我也偶尔会这样夜战。不过要注意自己的身体，不要耗光自己的精力。因为第二天上班你打瞌睡的时候，老板可不会认为是因为你昨晚熬夜了，是情有可原的。”

其实，加班并不是一件完全不好的事，有过编程经验的读者应该都有所体会，往往对一个问题进行解答，在夜深人静的时候或是睡前最容易有灵感。而且人的大脑越是到了晚上，越是冷静和清晰。

相反早上走进公司，一般需要一段时间来慢慢进入角色，而且白天会有很多琐碎的事情打扰。因此，偶尔趁着晚上环境好、脑子静可以事半功倍地进行项目开发，也不失为一种工作的乐趣。



提示 正因为如此，很多大公司都有严格的规章制度规定员工在上班时间不能随便打扰其他开发人员。因为，可能由于帮助解决你的一个小问题而打断了别人的思路，别人帮你解决问题花了 1 分钟节省了你 5 分钟的时间，恢复思路花了半小时，你总共浪费了公司 25 分钟。

2.4 莫学狗熊掰棒子

狗熊掰棒子的故事相信每位读者都听过，“每一次把棒子掰下来，都放到另一个胳肢窝里，然后用那只手去掰第二个棒子，结果第一个棒子就掉了，如此反复，到了最后，狗熊就只剩下手里刚掰下来的唯一一个棒子”。其实 IT 行业也存在很多有着“狗熊掰棒子”作风的人。

误区：从学生时代就是这样，做过的大小项目或是开发的小程序，都是做完就了事，删了或者是扔到一个以后不一定能找到的地方。自己在编程中遇到了问题，问同学，问网上高手，问老师，千辛万苦把问题解决了，也不做个心得笔记，只是享受了一下解决问题的喜悦罢了。

分析：这种人就是典型的具有“狗熊掰棒子”作风的人，好了伤疤忘了疼，说得一点都没错。程序开发是个脑力劳动，很多时候靠灵感想出来的东西过去一些时间后便会连自己都不能理解，更何况是自己出过问题的地方了。

2.4.1 做过的这辈子永远都不会忘吗

“蔡佳娃，你问了我这么多问题，我来问问你，大学这三年，你写过什么小程序，做过什么小项目没有啊？”

“小项目大概是没有做过，小程序倒是写过一些。”

“那你写过的东西都还在吗？删掉了吗？整理过吗？”

“整理？那东西需要整理吗？不过倒是没删，应该在电脑的某个文件夹中吧。”

“哎，那你写小程序的时候应该也会遇到一些问题吧？还记得都是些什么问题吗？”

“问题倒是遇到过，不过当时随便去网上问了问，解决了也就没再想了。”

“哎，看来我的小师弟几乎把该犯的错都犯了一遍啊。”

“啊，师兄，这也不对啊？”

“相当不对啊！你想想看，做过的东西这辈子永远都不会忘吗？犯过的错改正过一次永远都不会再犯吗？”

这句话说得很对，做过的项目和犯过的错，绝对不可能这辈子都记得。做项目的时候全靠大脑思考，而思考这个活动很难像在草稿纸上演算那样留下什么痕迹，有的时候自己都无法理解仅是几天之前所写的代码，更不用谈这辈子不忘了。

犯过的错误要想不再犯，其实也很难。在出错或遇到问题的时候，往往四处积极寻求指点和解答，但是一旦问题解决了，往往会长舒一口气，兴奋一下，然后继续编程。而问题的症结和解决方案却被渐渐地淡忘掉了。

“师兄，不过就算记不清，在遇到的时候应该还会很快想起来吧。”

“不要对自己的记忆抱有太多的幻想了。我有一次在网上浏览帖子，发现一个 Java 初学者问的一个问题。那个问题好熟悉，我想了半天，突然想起来这个问题在多年前曾经让我倍受折磨，我对当年的气急败坏至今仍印象深刻，只是我无论如何都记不起来自己是如何将那个问题放倒的了。”

“真的吗？记住了问题的模样，却忘了如何打败它的吗？”

“是真的，后来我把那个提问者的代码复制下来仔细研究了一下，终于想起来当年的解决方案了。真是不容易啊，我以为我再也想不起来了呢。”

“看来要想一辈子不忘还是不容易办到的啊！”

“不仅仅是遇到过的问题需要记住，自己曾经开发过的小程序小项目也最好都留着。”

“那些到了实际开发中还有用吗？”

自己曾经做过的小项目、小程序是很有价值的。小项目、小程序往往是对一类特定问题的解决方案，如果能够很好地保留，在再次遇到同类问题时不但可以借鉴，有时甚至可以直接修改加以使用。这样不但方便，还在很大程度上提高了工作效率。

很多小程序甚至代码片段虽然不长，但写起来并不容易，需要花很多工夫，比如一个特定算法的实现、一种特殊的界面开发方式、一个手机上传大文件的代码片段。如果这些自己已经做过的工作不能在今后的项目中使用、借鉴，而要花工夫重新发明车轮是十分可惜的。

“看来曾经做过的还是收集起来比较好啊。”

“的确是这样，而且当我再看那些写过的代码时，往往仿佛无字天书一般。一个原因是时间长了确实忘掉了开发的细节，第二就是当时我几乎没写注释。”

“但至少是自己写的吧？会忘得那么多吗？”

“到时候你就知道了，所以说养成写注释的习惯对于开发人员是多么重要啊！”

作为一名合格的开发人员，追求问题解答的执着和热情固然值得肯定，但是总结错误、总结问题、积累成果的开发习惯也是必须要培养的。阔步前进的同时思索沉淀一下自己的思想，或许会让前面的路更加好走。

不写注释或注释过于简练可能是 IT 新人最容易犯的错误之一，写注释有时候就好比将自己高妙、复杂的思想翻译成通俗文字。由于很短的一段表述复杂思想的代码有时需要很长的注释才能解释清楚，所以很多时候爱炫耀的开发人员都不太愿意做这种工作，觉得太耗费时间。

不过自己的思想有时候自己都会忘得一干二净，如何要求别人能看懂自己毫无注释的代码呢？如何在今后的开发中对项目进行维护和代码重用呢？

2.4.2 为自己维护一个小仓库

“蔡佳娃，所以说啦，以后在学习和编程当中，要不断地积累自己的成果，不要以为做过就算OK，将它们积累整理成一个小仓库，以后一定会大放光彩的。”

“恩，我回去试着整理整理以前做过的一些小程序，我当时也是没有加注释，顺便验证一下是不是自己全忘了，嘿嘿。”

“呵呵，你会对自己的忘性叹为观止的，不过也不要什么都往里放，一些太简单的东西比如冒泡排序法的源代码就不必往里放了，有找到的工夫都可以自己重写了。”

“是啊是啊，不过我做的东西这么少，没有多少拿得出手的啊！”

“慢慢来嘛，只要是自己觉得比较有用的东西都可以放进去，有时候 N 年以后再看自己的仓库，也会不由地对当年菜鸟的自己有微微的嘲笑，呵呵。”

“希望我早点进步到那个程度啊。”

“肯定会，皇天不负苦心人嘛！”

很多开发人员在初期都没有很好地注意整理自己的小仓库，使得有一些做过的工作没有产生应有的价值，当需要的时候又要再起炉灶，重新开发。这样不但浪费了自己的劳动成果，降低了工作效率，有时甚至再也找不到当时的灵感了，十分可惜。

古语说得好：“不积跬步，无以至千里；不积小流，无以成江海。”各位有志于成为 IT 精英、高级开发人员的读者一定不要犯上述人群犯过的错误，平时注意多积累、多整理，当用到时就事半功倍，效率胜常人一筹了。

“恩，要是早些听到你这席话就好了！我从今天开始就着手建立我的小仓库，看看多年后会不会被已成大器的自己嘲笑现下的愚笨，呵呵。”

“现在行动还不算晚，努力经营你的小仓库，小仓库也会时时回报你的。”

第二天，蔡佳娃拿着自己连夜搭建的小仓库给牛开复师兄检阅。

“师兄，你看我昨晚新搭建的小仓库怎么样？”（见图 2-2）



图 2-2 蔡佳娃小仓库图

“恩，做得还算可以，不过内容不太丰富啊！”

“啊？这还不算丰富吗？”

“声明一下啊，我说的小仓库可不仅仅是电脑上的几个源代码的文件夹，这个仓库是可以有多种形式的，可以是书面文档，可以是电子书籍或者印刷书籍，也可以是各种技术文章、帖子，或是一段视频。多媒体时代嘛，你的仓库也要灵活丰富一些。”

“哦，原来可以这样丰富啊！可惜我之前对这些根本就不关心，只好以后上网或者读书的时候遇到好的东西时再留心收集了。”

“恩，只要能找对方向猛冲，晚点出发也没有关系。这样吧，让你看看我的小仓库参考一下，这样你可以回去继续修葺一下你的小仓库。”（见图 2-3 牛开复小仓库图）

“恩，师兄你对我太好了！呵呵。”

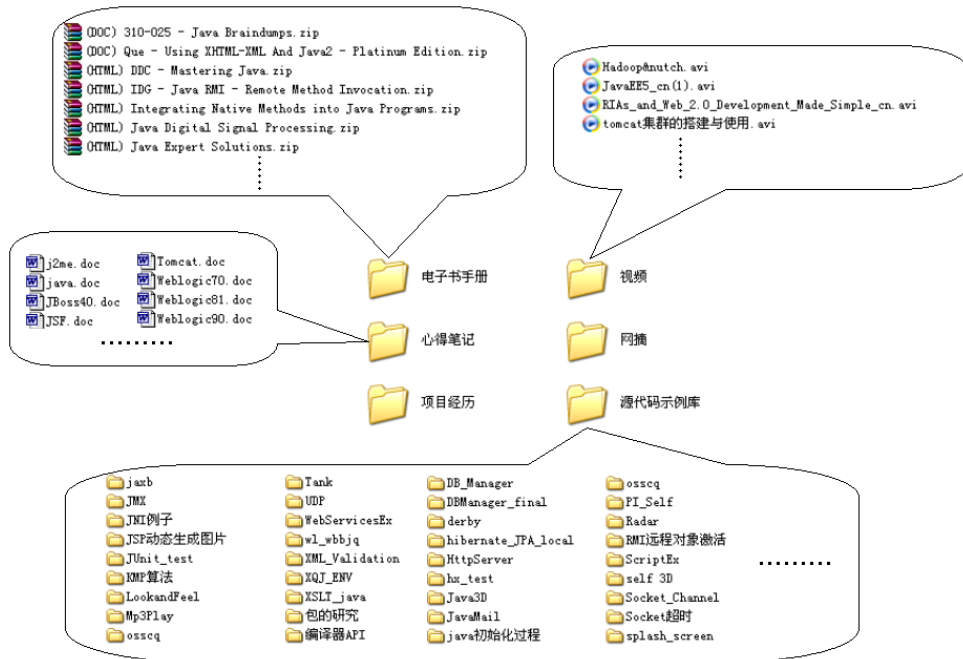


图 2-3 牛开复小仓库图

这个小仓库里装的，除了故事里说的那些自己做过的小项目、小程序的代码，还可以添加任何自己认为有用的代码、项目、文章、书籍等。

- 有些东西看了一遍，可能知识太深，自己能力还不到，不能理解透彻。可以将其存储起来等到自己能力到了，自然拨云见日，大叹“原来如此”。
- 有些东西自己压根就没时间看或是没有兴趣看，但是料到自己以后必定会用到的，比如看到一个 CSS 驱动的下拉菜单的代码，考虑到自己并不精通 CSS，但是经常进行 Java Web 的开发，可以直接存进小仓库而完全不用去细细研究，等到用的时候把案例调出来即可。
- 有些东西看了一遍，懂了，但是觉得很经典很高妙，对自己编程思想的提高有很大的帮助，绝对值得反复品味、感悟，这些东西也可以作为仓库的储备物。比如由国外大师 Erich Gamma、Richard Helm 等编写的《设计模式——可复用面向对象软件的基础》这本书，绝对值得每一位面向对象的开发人员时时反复品读。

2.4.3 多写开发心得

开发心得，或者称为编程笔记，对于很多新手来说可能很陌生，也很少有人会写。但是开发心得对于时刻检验自己、总结自己、完善自己有很大的帮助。同时，开发心得还是自己能力不断提高的见证。一个合格的开发人员会将自己的收获加到自己的开发心得中，作为一笔永久的财富。

“蔡佳娃，昨天回去搞了一晚上，你的小仓库修葺得怎么样了？”

“嘿嘿，还行，就是小点，我会慢慢丰富它的。”

“恩，不错，世上无难事，只怕有心人。不过光有一个小仓库是不够的呀，你还要养成写开发心得的良好习惯。”

“我们在学校进行专业课实习和做实验都需要写心得，项目开发也要写心得吗？”

“首先，你必须与学校里胡乱凑字数的心得划清界限，此心得非彼心得。你也可以把它叫做编程笔记，总之就是一些总结性的东西。”

“那都是总结什么的啊？”

很多新手都觉得编程笔记很难写，不知道什么样的内容应该放进去。其实不用在这些次要问题上花很多工夫研究，只要自己觉得有用的都可以总结出来，写进去。编程笔记里面主要记录了你在编程过程中的收获和思想。

就像前面章节中讨论过的那样，这些内容时间一长会忘得一干二净，因此需要记录下来以备自己查阅。另外注意，编程笔记主要是给自己看的，就相当于一个备忘录，不必对格式、行文等次要方面有过多严格的要求，花费太多时间。

“那师兄，开发心得可以和别人分享吗？对别人有价值吗？”

“开发心得不仅对自己有用，拿出去对其他人一样也是很有实用价值的。我们公司的一个前辈，将近十年的 Java 开发笔记有二百多页，我曾经跟他开玩笑说，你稍微整理、润色一下，丰富一些东西，就直接可以拿去当书卖呢！结果他照做了，那书后来卖得还挺火呢。”

“哈哈，看来 IT 界最重要的还是技术思想和经验啊！”


“那是当然，好好做你的编程笔记，师兄我等着你签名售书的那一天！”

也许初期开发心得的技术含量并没有小仓库高，但是编程笔记里面记录的都是自己曾经有过的闪光思想或齷齪错误。或许小仓库的精华可以为你指明行进的正确方向，而编程笔记中的点滴却可以让你少走些弯路，加快速度，提高效率，甚至开辟新的康庄大道。

总的来说，编程笔记中可以写进去的内容有：

- 闪光的思想：不管是在漫长还是短暂的编程生涯中，总会有某一时刻的想法让自己眼前一亮，或许这种想法并没有解决当时的问题，但是谁敢说不会在自己今后的项目中发挥重要作用呢？到时候就可以暗自感叹：“原来我早就达到过这种高度了啊！”
- 错误和问题的记录：在开发过程中，肯定会遇到难以逾越的问题，或是容易犯的错误。问题解决了可以将其记录下来，等到再次遇到的时候就避免了绞尽脑汁也无法想起来的痛苦了；错误只有犯得多了才会记住，不过如果及时记录下来的话，可以让自己少犯几次。

- 经验的浪花：这或许是开发心得中最精华的部分了，编程知识如果是沙子，那么编程的经验无异于沙子中的黄金了。这里的经验并不是指开发过多少项目，而是在开发过程中通过实践得出的一些规则以及解决具体问题的方法。这些内容是哪一本教科书或任何一种课堂无法直接传授的。

 **提示** 笔者作为有多年开发经验的开发人员深深感受到，有很多经验在你没有接触过具体问题时就别人与你分享，你也很难理解。大部分情况下你还是会去犯那些错误，只有自己从实践中总结出来的经验才是最有价值的，读者朋友们平时一定要注意多总结。

2.5 本章小结

已经知道了暗礁在哪里，对于一个打算成为伟大航海家的人来说，所欠缺的或许就剩下勇敢地将船驶向大海了。或许了解暗礁在哪里，并没有学习如何开船来得爽快，但却会为航海增添一份强有力的保障。“天行健，君子以自强不息”，IT 这片神秘而又汹涌的大海，就等着你来遨游。



Java 程序员 职场全攻略

——从小工到专家

吴亚峰 著

懵懂菜鸟如何才能成长为一名成功的开发人员？

如何在“血雨腥风”的求职大战中脱颖而出？

职场中只要技术过硬就可以吃遍天下吗？


是学招式重要，还是练内功重要？

掌握了哪些技能，才算是真正的高手？

.....

资深技术专家，娓娓道来。



 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

第 3 章 下山之路——有备无患

如果把 IT 这个新世界比作江湖的话，那么每个人在进入江湖之前，都必须在深山中苦练武艺。因为江湖险恶，稍有不慎就会迷失江湖。只有准备充分，才能让自己的下山之路更加平坦，让自己顺利地从小卒成长为江湖大侠。

3.1 从学生升级到开发人员

对于本书的大部分读者来说，进入江湖之前练功的深山，就是学校。如何从学校里舞刀弄枪的习武小子，成长为可以独当一面的江湖菜鸟，这个过程是每个人都必须经历的，也是很多人都颇为困惑的。本节将向读者介绍如何让自己从学生升级为合格的开发人员。

“师兄啊，当初你是怎么从整天上下课的学生直接过渡到每天上下班的开发人员的啊？”

“过渡？我没感觉啊？就是该干什么干什么呗！”

“可是我总觉得自己离一名开发人员还有很大的鸿沟呢？”

“鸿沟？没那么严重吧，看来我还得给我的小师弟做做岗前辅导了，呵呵。”

“其实就是觉得自己离着一个合格的开发人员还有些距离。师兄你指点指点我，让我有备无患吧！”

“恩，我明白你的意思了。好，让我来指导你做下山之前的准备。”

3.1.1 学校给了你什么

学校，或许是每个人步入职场之前呆得最久的地方了。不管学生对学校的态度是好是坏，是爱是恨，学校对于学生今后的发展都有很深的影响。但是，究竟这个占据我们生命至少五分之一的求学生涯给了我们什么呢？

“师兄，你看我都上大三了，怎么总觉得自己还是拿不出手呢？”

“很多时候是由于你对自己和 IT 职场不够了解的原因，不过我们之前已经探讨过很多这方面的问题了，从 IT 职场的揭秘，到 IT 职场的误区预报，你不应该这么迷茫啊？”

“这个我倒是懂，只是我不太明白大学这四年，包括之前上学这么多年学的东西够不够用啊？”

“原来你是对学校教你的东西不自信啊！看来还是让我这个师兄出马，助你提升一下功力吧！”

对于现在的大学教育，抛开那些只占全部大学一成的全国重点高等学府，可能所有人都会承认学校教育和社会需求出现了或多或少的脱节，不管他是学生、老师，还是校长。学校的知识或者是过多地落后于象牙塔之外，或者是理论讲了一大堆，一到实践就基本瘫痪。

但是脱节并不代表相去千里，不意味着毫不相关。纵观当今 IT 行业的人才需求，学校教育仍然是每个职场新手步入社会之前的必修课，这主要包含以下两个方面。

- 学校知识是基础

- 知识以外的收获

1. 学校知识是基础

“师兄，我觉得大学这四年在学校里呆着，有些浪费。”

“嗯？为什么这么说呢？”

“学校里开的很多课都没有用嘛，什么高等数学、数值分析、大学物理、复变函数之类的，根本和 IT 这个行业无关嘛！”

“哎，我的小弟啊，你如此不知书滋味，小心书到用时方恨少啊！在你眼中什么才是有用的呢？数据结构？网络原理？操作系统之类的？”

“不是吗？学就学点实在的，拿出来就能用上手的知识。”

“哈哈，我只能再次鄙视一下你的目光短浅了，不过也不能全怨你，现在的社会太急功近利了。你这种实用技术为上的态度我在大学里也有过，我可以告诉你我现在有多后悔当初没学好那些基础课，尤其是与数学相关的课程！”

不光是故事中的蔡佳娃，现实中很多大学生对于学校开设的基础课都不以为然，反感至深。比方说数学，计算机的指导思想和基础就是数学，早期的计算机科学家有很多都是数学家出身，而现在的高级计算机学者中也有很多人都在研究数学理论以提升算法效率。

需要说明的是，像数学、物理等基础课程中所讲的理论都是从长期的实践中经过总结提炼出来的，学习这些理论的目的就是在需要的时候将其再次应用到实践中。这就可以在很大程度上提高实践的效率，避免重新发明车轮，或者根本就发明不出车轮。

而 IT 又是个面向各个行业的服务性行业，客户提出的需求可不一定只限于 IT 这个专业，不是几个循环和分支判断就可以堆砌出所有实际应用的，因此能够掌握这些基础知识对于以后的快速、高效发展就大有裨益了。

“啊？原来那些东西还真的有用啊？”

“到时候你就知道了，我记得我在工作的时候不止一次回头翻看离散数学、概率统计的课本呢！那些你所说的没用的课，你是不是基本都没学好啊？”

“呃，也不是啦，只是兴趣不大，旨在不挂而已。其他还行，大学物理比较差，勉强及格。”

“看看，我记得你曾经说过对游戏开发挺感兴趣，那你知道游戏开发中渲染一个 3D 场景需要做什么工作吗？”

“不是直接写代码吗？”

“很多地方都要解薛定谔方程！估计你连这个名词都不记得了吧！”

学校里每一门学科其实都是面向应用的，只是在教学的过程中或者是课程的安排有问题，或者是老师能力的缘故，只注重了知识的传授，忽略了将其融会贯通，结合起来联系实践。因此有些课程看起来似乎太离谱，几百年也不会再用到。

很多人在这种情况下表现出来的逆来顺受会让人很难理解，而有些人也会和故事里的蔡佳娃想的那样，既然课程不好，或者老师不好，干嘛费心学呢？不如学点自己认为有用的东西。但这样做的结果最有可能是丢了西瓜捡芝麻，捡来后却发现还是变质的芝麻。

“师兄你说得对。那些课程我是没有好好听，当时我想更好地利用时间，所以都会自己去看一些有用的书。那段时间我从图书馆借了好多书呢。”

“好，那我问你，借了那么多本书，从头到尾看完的有吗？有几本？”

“这个嘛，呃，好像没有吧，都是看一大半就被什么事情打断了，或者看到后面内容太深看不懂，或者是该考试了，得抓紧复习了。”

“所以说嘛，有的时候到为了抓紧时间双管齐下，认真学习课本知识的同时又大力钻研自己认为更有用的技术，但由于计划混乱，最后的结果往往是兵败两地。你应该好好利用在校学习的时间，每一门课都或多或少有它存在的价值。你可以抱怨学校对你不负责任，老师对你不负责任，但是不可以自己不对自己负责任啊！”

“嗯，师兄你说得太对了。我以后得好好规划一下自己的时间，让自己不再竹篮打水一场空。”

“对喽！学校的知识不是万能的，但不会是万万不能的呀！”

在学校所学的都是基础，虽然效果不像学一门实实在在的技术那样立竿见影，但是学校的知识会在潜移默化中提高一个人的思维能力和理论功底。比方说数学，数学是解决实际问题的一种抽象思维，现实中很多问题都可以理解为数学模型来解决。所以说数学是思维的体操，一点没错。

例如游戏中要开发水波纹的场景，就要用到与波方程相关的数学物理知识；应用中要优化一组解，就要用到最小二乘法等。如果在学校学习时不重视，等到要用时不但不会，甚至都忘记它们的存在了，那就很难胜任工作了。

在这里提醒那些完全抛弃学校教育而只靠自学的同学，自学精神的确可嘉，也很有必要，但是绝不可以彻底否定学校的教育。不是学校给你的知识不好，而是你没有用心去学。或许课本知识没有学习一门技术有激情，但是技多不压身，可以不精通，但是不能听都没听过，没准哪天会因为你比其他人更了解薛定谔方程或自动机原理从而会脱颖而出。

2. 知识以外的收获

“师兄，听你这么一说，我今后还是踏踏实实地学些课堂知识，要不然内功修为不够啊！”

“这样很好，不过说实话，学校的课程里面还真是有一些枯燥无用的，或者是彻底过时的，这些课程你自然可以抱着六十分万岁的心态来对待咯。”

“是是，我还是能分清什么课程是可以放弃的。”

“那样就好，其实学校除了给予你出来混所必要的一些基础知识以外，更重要的是养成固定的、良好的习惯。”

“习惯！这个和以后的工作有关系吗？”

“当然有了，我所指的习惯包括生活、学习、思考、做人等各方面，大学生活中养成的习惯很多时候会跟随你一辈子呢。”

“啊，有那么大的影响吗？”

大学是走向社会的最后一个准备阶段，在大学的四年学习生活中，每个人都会在性格爱好、学习思考方式、待人接物甚至生物钟方面形成一些固定化的模式，这些就是习惯。习惯一般是最不容易被注意到的，很多时候习惯一旦养成就很难改变。

习惯成自然，或许这句老话已经过时很多年，但是道理却一直都是正确的。很多职场新手有

时不是败在自己技术不行、脑子迟钝、经验不足上，而是其或大或小的一两个沿袭自学校的坏习惯使自己自断神兵。

“我给你举个例子吧，我们公司前些日子新进来一个大学毕业生，技术能力不错，脑子也够灵活，可办事总是拖拉，主管分给的任务他总是最后一个完成。”

“哦，那是为什么啊？”

“主管也很纳闷，本来很聪明能干的人，不应该表现得这么拖拉啊。于是主管叫我观察一下这个新员工，结果我发现了他的秘密。”

“什么秘密啊？”

“这个人的确非常有能力，我见过他写的代码，很规范很华丽。可是我发现他无法让自己的注意力集中时间超过 30 分钟，30 分钟后，他就有些开小差，查阅一下邮件，接杯水，浏览一下其他网页等，就这样杂乱无章地玩一会，然后再继续工作。”

“所以他把时间都给浪费了？”

“是啊，然后我就问他为什么会这样。他说他在大学的时候就是这样，刚刚开始学编程的时候，每次写完一段难度很高的代码，就很有成就感，就想奖励一下自己。而且他宿舍的其他人一般都在玩游戏，他也受一些影响。”

“啊，看来学校里面养成的习惯真会带到工作中啊！”

“是啊，后来这个同事花了很长时间才跟上我们的工作节奏，要不是因为他有点天分，估计早把他辞掉了。”

除了故事中提到的那个无法长时间集中注意力的坏习惯之外，还有一些坏习惯需要避免。比如一看书就头疼打瞌睡、大早起来特别困等。都说 21 天可以养成一个习惯，然后用 90 天的时间使之趋于稳定，不知道改正一个坏习惯是怎样的流程，估计比养成习惯难 N 倍。

同时，在学校的学习中，也会逐步建立起比较专业的思考问题方式，并逐渐将自己的思维模式固定下来。经过大学的历练，不仅知识储备上必须达到要求，个人的学习工作习惯和思维方式也应该提升到能胜任合格的开发人员的水平。

“师兄，听你这么一说，看来学校对我们的知识和思维能力的潜在影响还真是不小啊！”

“是啊，另外，学校虽然是个教书育人的地方，但是现在的大学也越来越像个社会，在大学的几年时光，或多或少地也教会了我们一些为人处世的道理。”

“恩，师兄你说得没错。有时候如何处理好跟其他人的关系，也是比较考验人的。”

“那是当然，其实到了社会上也大概就是这个样，同事就像同学，既要团结友爱，又要互相竞争；老师就像上级，既有敬佩听话，又有怀疑不服。虽然学校远没有社会上那么严重，但是至少在学校能让你提前感受一下这个气氛。”

“是啊，师兄，说得很有道理。做人难啊，呵呵。”

“做人才更难，做个好的 IT 人才，那是难上加难啊，哈哈。当然了，学校也没有如此地不近人情，它主要还是提升我们各方面素质的地方。”

从学校获得知识的同时，也收获了很多习惯，养成了一些基本固定的思维方式和做事的准则。学校虽然不能为你以后的工作做百分之百的保证，但是如果你没有荒废学校的教育，那么就算万

丈高楼倒下了，至少地基还是在的。

总结一下，从学校中应该获得的有以下几点：

- 基础知识，包括专业知识和其他领域的知识；
- 学习和工作的习惯；
- 思考问题及做事的方式；
- 为人处世的一些学问。

3.1.2 咱们还缺啥

学校教育走向职场的基础，这一点是可以肯定的，但是只有学校的教育够不够，这就见仁见智了。IT 行业是个技术发展、更新很快的行业，尤其是软件开发方面，新的技术层出不穷。因此，学校这个比较庞大的列车在处理这些新技术时偶而就会显得有些力不从心了。

“师兄，上次你跟我说不要小看了学校教育，要让自己的基础知识尽量扎实，但是其实光靠学校教育是远远不够的吧？”

“就我们 IT 行业来说，的确直接从学校教育走出来的学生离企业需要的人才还是有一些差距的，脱节现象比较严重。企业需要的大都是不管材质如何拿来就能用的成品，而学校培养出来的大部分都是材质不错但是欠缺雕琢的半成品。”

“所以在学校知识之外，我们还得掌握一些必备的专业知识来应付求职之路。”

实际上，如果学校开设的课程还算合理有用，不是完全的落俗套、做表面文章，那么把学校的知识彻底掌握，是完全可以找到一份比较满意的工作的。因为具有扎实的基础知识的人才相较于那些故事中提到的只注重表面技术的技巧型人才，同样也是有一定人气的。

但是在一般情况下，从学校顺利毕业后还是很难按照自己的理想满意就业的。这里面的原因很多，学校、社会有责任，但更多还是在个人。这也是为什么如今的 IT 培训和认证考试如此流行和受欢迎的部分原因，认证将会在后面为读者介绍。

“正解，但从掌握知识这方面来说，大学生毕业时应该至少精通一门编程语言，会用其编写简单应用，熟悉数据结构和算法方面的知识，以及计算机网络和数据库方面的知识。”

“光看那些程度词‘精通’、‘熟悉’就知道企业对于人才的要求还是蛮高的啊。”

“对啊，所以不能再像以前应付学校那样来应付公司啦。我刚说的这些知识，或者是学校里没有开设相关课程，更多的是有相关课程但是学了不够用，或不会用。”

“师兄，你说得太正确了！我对这个深有感触，曾经学了 C 语言后觉得自己很牛，然后去网上的论坛里瞎逛，发现人家讨论的问题我压根都没听过。”

“是啊，种种原因吧，学校没有办法让我们进行最充分的岗前训练。我们在踏入职场之前最好是在学校知识的基础上深化一下自己的知识，这样才能顺利搭上 IT 的特快列车。”

学校教育是面向大众的，因此要兼顾不同水平的学生，所开的课程就不可能过于高深。况且学校的教学水平也是有高有低，对于新技术的反应也有所不同。所以很多时候学校教育和企业需求之间的鸿沟需要求职者自己来填平。

企业对人才所具备的知识需求主要有以下几个方面：

- 编程语言方面，需要熟练使用一些辅助语言，至少精通一门主编程语言。这里的辅助语言主要指那些通用性高被广泛应用的语言，如 SQL、HTML、CSS 等，这些语言一般功能比较单一（如 SQL 只面向数据库处理），但是却具有不可替代性，所以必须熟练使用。而需要精通的主编程语言则是那些功能强大的高级语言，如 Java、C/C++ 等，这些语言将会成为进入职场后开发时的主要语言，所以必须毫不含糊地彻底拿下。
- 数据结构和算法以及设计模式，这些是永远都不会过时的内容。武侠小说中经常会提到的“万剑归宗”的“宗”，在计算机技术中应该就是算法和设计模式了。编程语言只是一种工具，而算法和设计模式则是语言无关的。
- 计算机网络原理和通信协议，随着网络技术的发展，不具有网络连接功能的软件已经基本退出了历史舞台。因此要想做一名合格的开发人员，网络原理的知识必须了解，而通信协议的内容则必须熟悉，并能够根据其原理设计开发自己的网络应用协议。

“师兄，听你这么一说，我们离公司招聘的需求还是差得不少呢。”

“呵呵，所以说剩下的这两年还是不能放松啊！”

“那是必须的啊，不过师兄，提升自己、弥补学校教育缺憾的途径是什么呢？”

“一般来说，有三种方法：自学、网上获取资料、参加培训。首先说说自学，自学算是成本最低的一种提升方式，但也是一种水滴石穿的过程，因为自学全靠自己的领悟，而且自学需要对自己的约束能力很强。自学能力也是一种很重要的能力。”

其实一个人的自学能力一般也是从大学开始培养起来的，进入大学后学生需要自己主动地学习，家长和老师一般不会太多的干预。而进入职场后无论是忙中偷闲努力缩小和别人的差距，还是快马加鞭大胆尝试新技术，靠得都是自学能力。

而且最新的技术一般是没有完善的教学体系和资料的，也很难从别人那里学习，能否学好用好新技术主要靠的就是自学能力，因此读者朋友们一定要注意自学能力的培养。

“说得没错啊师兄，我的自学能力就不够好唉。”

“很多时候自学能力不强是因为压力不够大，你慢慢就知道抓紧了。一般情况下自学都和第二种方式联系起来，就是网上获取资料。”

“恩，我想也是，现在网络上什么都有，肯定特方便。”

“对啊，网上获取资料包括在社区论坛里面提问讨论，从网上获取各种形式的电子教程和源代码，了解相关技术的最新消息、获取各种技术支持如开发环境和源代码等。”

“是啊，我的电脑里还有很多视频教程呢。”

“网络上的资源相当丰富，而且新技术的介绍也是网上快于书面，所以不利用这个资源肯定是个浪费。不过网上资源和自学有个比较大的缺陷，就是交互能力不够强。”

“交互能力？”

“对，就是在学习的过程中遇到难以逾越的问题，在学校教育的模式下会去向老师询问，就是和高手交互，而自学和网上获取资源由于没有老师，所以这方面显得有些不足。”

“可是，可以在网上提问题啊！现在这种网站很多啊！”

现在这种问答形式互联网服务很流行，而且查询起来也非常快捷，但是有些时候，在网上提

的问题很少会有让人满意的回答。有时获得了满意回答后，想进一步讨论也很难。

最一般的情况是，很多人的回答要么简练无比不知所云，要么答非所问一头雾水，偶尔有一两个回答很对路，进一步追问追问，结果人家再也没回过帖。不可否认的是，网络上不乏高手，但是能找到一个真正能够排忧解难的人恐怕很难。

而且要注意的是，网上能有人给出答案的问题一般都是对菜鸟特别有用。比如帮助修改几行源代码、重新配置服务器等，这些对成手而言都不是问题。真正想要在网上根据自己的需要讨论复杂一些的思想性问题是很难的，这点笔者深有体会。

“师兄，听你这么一说，想想我曾经在网上苦等一个满意答案的悲惨经历，网上问题解答的方式还真不是非常有效的呢。”

“从网上获取一些小问题的解答是很方便的，比如 Tomcat 服务器的配置等，所以还是要对网上的资源敞开怀抱的。最后再说说培训这个途径。”

“这个我听说过，培训就是花钱去学技术。”

“是这么回事，不过培训相比前两个途径的优势就是交互能力强，因为培训一般是传统的老师和学生面授的形式，所以在解答问题等方面是很有优势的。”

“是啊，比网上更快，呵呵。”

“同时培训比较符合学生的学习习惯，毕竟人们在网上学习才几年，而在教室里上课的历史可是很长了。再一个原因就是参加培训一般要花上一笔可观的学费，这也应对了我刚才说的有压力自学能力才会更强，因此参加培训的学员会比其他人自学起来更有动力。”

在真正的学习中，最好还是把三者或者至少前两者有效地结合起来，尽量提高自己的水平，使自己在学校知识的基础之上补漏拾遗，更上一层楼。

3.1.3 经验，还是经验

搞定了学校的基础知识，又提升了自己的能力以达到标准，最后来讲一讲让很多即将毕业的大学生谈之饮恨话题：项目经验。很多公司在面试的时候基本上都会问到的问题就是：谈谈你曾经做过的一个项目。所以为了让自己成功跨入开发人员的行列，经验还是必须要有的。

“师兄，经过了学校的学习和自己的努力，应该可以满足一个招聘单位对求职者的要求了吧？”

“恩，知识上是差不多了，实践方面还是需要加强，比如要有一定的项目经验。”

“啊？最后还是要求经验啊？我们刚刚毕业，哪来的经验啊！”

“这也是用人单位没办法的事，用人单位其实只是想要一个技术好、有能力的人，可是技术和能力这东西短期内也不大容易看得出来。所以加个项目经验的条件，自然会让其招进来的员工更加保险一些。”

“是啊，现在职场竞争这么大，无经验无门路啊！做大学生难啊！”

现在可不是抚琴慨叹、怨天尤人的时候。说得残酷些，那些零经验进入职场的人，在学校考零分和满分是没有差异的，因为他们都将面临淘汰的可能，就好比去开发房地产，地皮买好了，打个地基，插几个桩子就开始卖房收全款，那样不现实的做法肯定不会有人买账。

因此，在校大学生朋友们虽然在踏出校门前很难有工作经验，但是通过自己的努力获取一些

项目经验还是十分有必要的。获得经验的大致途径后面会介绍，这里读者不必担忧。

“是啊，这个说得倒是没错，我之前照着一些书上的内容自己写程序，书上的都看明白了，自己真正写的时候就会蹦出这样那样的错误，有时候跟书上的东西一模一样，但是到最后还是会发现忽略掉的细节和错误。”

“‘纸上得来终觉浅，绝知此事要躬行’，IT 行业对实践能力的要求是相当高啊，所以说为了自己保险起见，你也要有一些项目经验才行。”

有句古话说得好，“腹有诗书气自华”，真正有能力的人无论走到哪里，都像黑夜中的萤火虫一样明亮显眼。但是还有一句古话：“千里马常有，而伯乐不常有”。虽然你很闪亮，但是有可能没有碰到赏识你的人，所以项目经验这个实实在在的东西就显得非常有必要了。

这里有必要区别一下知识与技能之间的不同，因为掌握了知识并不代表具有了技能。知识和技能的转换关系可以用如下的方程式来表示，如图 3-1 所示。

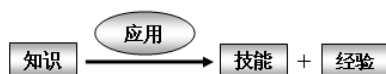


图 3-1 知识与技能转化图

图 3-1 的含义就是知识在反复应用的条件下会生成技能和经验，很多求职者都停留在方程式的左边，而用人单位往往需要的是右边的技能。知识在转化为技能的同时也会丰富一个人的经验，而相对于技能而言，一个人的经验如何往往更容易检测出来。因此很多情况下，根据求职者的经验介绍，用人单位就可以对其技术能力的水平做一个大致的判断。

“师兄，那我们这些人应该怎么混出一些经验呢？”

“每个人提高自己水平、积累经验的方式都不一样，我来讲讲当年的经历吧！”

“恩，我们这种菜鸟最喜欢的就是举例子。”

“我在大三下半学期的时候，觉得自己知识储备得差不多了。当时没有人可以投奔，也没有人给我任务，所以我就花时间给自己出了个题目让自己做。”

“那是什么题目啊？”

“一个图书馆的信息管理系统，做完那个模拟项目的确让自己进步了很多。后来学校里一位老师接到了一个项目，需要帮手，接着我就去了。老师因为看到我之前做的那个图书馆信息管理系统的项目还可以，就留下我了。”

“这算是师兄你最初的项目经验吧？”

“差不多吧，老师分了一些任务给我做，之后我又帮着那位老师做了几个项目。在大四下学期开始的时候，我已经在一家老师推荐的公司断断续续实习了将近一个学期了。”

“是啊，这么一来师兄你的境界和那些一直闷在学校里的同学可就大不一样了啊。”

“呵呵，这些你可以听听，具体怎样让自己的知识转化为技能和经验，还是要靠自己的努力呀，记住你曾经说的话：无经验无门路。”

“恩，师兄，我会找到自己的提升道路的。”

从学生到开发人员的升级之路，说起来很简单，但是每一步做起来都不容易，找工作最好的境界就是胸有成竹，其实成竹在胸的就是技艺精湛、经验十足的底气。总结一下这条下山之路首

先要完成的任务有如下三个：

- 学校获得的知识或其他东西是自己进入职场的“地基”，没有地基的房子固然不一定会立即倒塌，但必定是鲜有人问津。
- 不断提高自己以满足不断变化的企业需求是进入职场的“发动机”，说到底主动权还是在自己，能不能让自己发动起来就要看自己的干劲了。
- 多写多练注重实践经验的积累是进入职场的“助推器”，既然是助推器，貌似没有发动机重要，但是却决定着开发人员这辆车的速度与能效。

3.2 为自己定下目标

对于每个将要踏入 IT 这块战场的菜鸟来说，为自己定下一个目标并努力去实现将会是一个非常好的进步方式。如果把一个人的职业生涯比作一个巨大项目的话，制定不同的目标就是在完成不同的模块，只是这个项目只有自己能完成，而且没有别人设定的进度限制。

3.2.1 目标的意义

“蔡佳娃，今天我们来谈一个比较崇高的话题。”

“什么崇高的话题啊？”

“目标，在进入职场前要为自己定下目标，并且努力去实现它。”

“哦，这话题也不算崇高嘛。”

“目标是一种务实的说法，我如果说理想是不是就有崇高的感觉了？呵呵，闲话少叙，你有没有为自己立下过什么目标啊？”

“这个怎么说呢，目标倒是定过，不过我没觉出来定不定下目标有什么区别啊？”

“当然有区别了！哦，我明白了，看来和你讨论目标之前还得先奠定一个基础。”

“啊？什么基础啊？”

“就是目标是用来干什么的啊？”

如果给别人解释目标的含义肯定很少有人喜欢听，因为目标的含义很简单，就是一个人行动的方向和期望达到的结果。目标有长有短，有大有小，我们时时刻刻都在为着某个目标努力，长到一个终身奋斗的理想，短到一两个小时的工作计划；大到引领一番变革的风浪，小到给自己一个满足的欣慰。

作为一个即将投入 IT 浪涛中的新手，最好为自己定下一个目标，而且必须是个不能太小且长期的目标，这样才能找到前进的方向，不至于在狂风颠簸中迷失自己。

“所以啊，蔡佳娃，你要为自己定下一个长远的目标，这样你用起功才会无怨无悔，甘之若饴啊！否则每天只停留在自己的小计划中，慢慢也就会失去工作的激情。”

“恩，师兄说得对，没有目标就跟没有舵的船一样，迟早会迷失的。”

“正解，纵观历史中风起云涌的各路英雄，偶然成功的人有，但更多的还是定下长远的目标并一步一步按着自己的雄心走出来的。”

“是啊，就像武侠小说中写的一样，世外高人、怪侠异人层出不穷，但是统领江湖、长久不衰

的，还是那些由小及大、逐步发展壮大的名门正派。”

“呵呵，你武侠小说看多了吧。下面我给你讲一个靠远大目标赢得长久胜利的英雄故事，故事的主人公叫做帕特里克·J·麦戈文。”

帕特里克·J·麦戈文这个名字或许不像比尔·盖茨和史蒂夫·乔布斯那样鼎鼎大名、如雷贯耳，但是或许大家读过一本杂志叫《计算机世界》，《计算机世界》的创办者就是麦戈文。正像比尔·盖茨打造了微软帝国一样，麦戈文作为 IT 界的传媒大亨，打造了 IDG 帝国。

如果读者有兴趣，也可以到图书馆找几本《计算机世界》来读一读。读过就会发现，《计算机世界》还真是 IT 界很有影响力的杂志，从杂志中可以了解很多行业的最新动向与发展趋势，这对我们 IT 人来说是很帮助的。

“麦戈文首先是一名资深的 IT 媒体记者，与其他 IT 媒体经营理念不同的是，麦戈文创办 IDG（国际数据集团）的时候，就将公司的发展目标定为面向全球的具有国际性质的 IT 信息提供商，这也是其主打刊物《计算机世界》得名的部分原因。”

“树立了这样的目标，麦戈文便开始了自己创办世界级的 IT 舆论阵地的王朝之路，当时的 IT 杂志都是靠广告费生存，谁给的钱多就为谁说好话。麦戈文看到了这种商业模式的弊端，在 1967 年创办了《计算机世界》杂志，由此标志着这个帝国的崛起。”

“凭借着他对市场的敏锐嗅觉和资深的 IT 媒体记者经历，《计算机世界》迅速成为全美最受欢迎的 IT 杂志，在公司成长到一定规模之后，功成名就的麦戈文马不停蹄地继续实现着他的宏伟目标，在 1974 年将《计算机世界》带进了欧洲。”

“随着 IDG 公司在 IT 媒体界的舆论影响力与日俱增，麦戈文将自己的出版物发行到了全球 100 多个国家，推出了 20 多种语言版本，分公司遍布世界各地。”

与此同时，很多当时和麦戈文的《计算机世界》同时起家或者更早的 IT 媒体，由于没有像麦戈文那样早早地定下全球战略，都在 IDG 的巨大光环下或慢慢退出，或被收购。而麦戈文一手创建的 IDG 由于正确的目标指引，成为当今 IT 世界最有影响力的媒体平台。

“蔡娃娃，听了这个故事，你是不是有点想法了？”

“恩，师兄，虽然我做不到麦戈文那样的 IT 媒体皇帝，但是我也会为自己立下一个目标，努力前进，或许一个目标一个目标的实现，有一天我也会发现，我成功了。”

“恩，你有这样的想法非常好，师兄期待着看到你的进步！”

目标促使一个人将准备要做的事情细细规划并实现，一个人只有明确了目标才能找到前进的动力，才能为自己付出的异于常人的努力找一个合理的借口，才能让自己在别人的不理解和迎面而来的困难面前不为所动，持续前进。

3.2.2 树立目标的学问

前一小节提到了目标的重要性，但是很多时候导致失败的并不是没有目标，而是目标不够合理，或者目标太多，又或者目标频繁变动。因此，如何立下正确的目标也是个马虎不得的重要过程。

“师兄，上次你只是说了定目标的重要性，那么定目标时需要注意的问题是什么啊？”

“首先，目标要恰当，要适合自己，不可盲目自大，也不可妄自菲薄。要首先对自己有正确的认识，才可以立下合适的目标。”

“恩，我估计我有点不太自信，可能立的目标会比较屈才。”

“目标定得过小，你的才能就无法彻底发挥；而目标定得过于飘渺，有可能会让自己的前进道路荆棘遍地而困难重重，最终无法实现。”

“所以要尽量避免做一个眼高手低的人是吧？”

“正解，有些人的目标过于宏伟，结果在向着目标前进的过程中总是碰壁，碰来碰去，不仅鼻子扁了，意志也被消磨掉了，离成功也就会渐行渐远。”

目标定小了，实现起来会容易一些，也可能会增添一些信心，但是也有可能变得夜郎自大，认为所谓 IT 精英也“不过如此”；而目标定大了，就有些不切实际，往往会在不可避免的挫折中丧失信心，以失败收尾。

在实现目标的过程中，发现错了，不要对自己太容易懈怠和放弃，也不能硬着头皮死扛，而是要立刻进行修正，以免自己的努力付诸东流。

有些人在立下目标的时候只是逞一时之快，感受一下自己的豪情万丈和一腔热血就了事，之后便又回到原来的老样子，等到哪天又受了刺激再来一次振臂高呼，好像目标只是一首励志歌曲，可以唱在嘴里，却无法做出来。

“另外给自己定目标时还要注意的是不要频繁地变动当前的工作中心，爱迪生不是说过嘛，‘比起有一千个有创意的想法而一个都不做，我更喜欢只有一个创意而努力去实现的人。’所以说目标不管大小，不随意改变也是很重要的，贵在坚持嘛！”

“是啊，我们的导师也经常说不管大家选择考研还是就业，一旦选择了，不要变就行。”

“说得没错，有些人就喜欢变来变去，今天看了李开复的演讲视频就拍案而起满腔豪情地立志要做个有为青年。刚刚发愤图强了几天，四六级考试又迫在眉睫了，只好先救近火，全身心地去背单词，考过四六级之后又会有新的事情加塞进来，如此反复……”

“师兄你分析得很透彻啊！”

“呵呵，下面我再给你讲个 IT 界伟大思想家的故事吧。他的名字叫做道格·恩格尔巴特，不知道你听过没？”

“没有唉，我这个人太孤陋寡闻了。”

其实也不能怪蔡佳娃孤陋寡闻，道格·恩格尔巴特或许是 IT 历史上最容易被忽视的一位思想家和发明家了。恩格尔巴特在几十年前提出的非常有预见性的技术和观点都不被当时的主流计算机学者所接受，而现在计算机能有如此友好的人机交互界面全都是拜他的坚韧努力所赐。

关于鼠标的发明者，很多人只知道是一个海军军官，其他的便一无所知。其实恩格尔巴特就是鼠标的发明者。当我们今天在电脑前自如地使用鼠标的时候，却不知道它的发明者为此经历了多少的挫败和打击。而如今他对于计算机发展的设想都广泛地应用在每个人的生活中，而不仅仅是鼠标。

“恩格尔巴特接触计算机技术是在他 25 岁那年。之前他一直在美国海军服役做一名雷达兵，退役后他做了一名电气工程师。当时计算机的输入输出靠的全是穿孔卡片，而天才的恩格尔巴特

却开始有了为其配置输出屏幕等设备以增强人机交互能力的超前想法。”

“恩格尔巴特为了他的目标开始了奋斗，他先在加州大学伯克利分校攻读博士学位，而后又在斯坦福研究所工作，还到社会上去寻找能支持他的合作伙伴。但是所有他遇到的人都认为他的想法不可理喻，也没有几个人能听懂他的理论。”

“那不跟爱因斯坦当年提出相对论一样吗？”

“的确是这样，与爱因斯坦一样恩格尔巴特依然没有放弃自己的梦想，终于美国军方开始对他人机交互的研究感兴趣，并给了他一笔经费做研究。在进行研究的初期，恩格尔巴特也是孤军奋战，直到后来他的文章逐渐引起人们的注意，不断有人为他的研究提供资金，他才开始真正地铸造自己的梦想。”

“1968 年恩格尔巴特终于等来了机会，在一次电脑会议上，他向观众展示了他的研究成果，其中就包括鼠标的亮相。这次会议绝对值得被载入 IT 的史册，因为恩格尔巴特在那次会议上演示的人机交互思想与技术，在从那之后的几十年里都如火如荼地发展起来了。”

“或许是天妒英才，正当恩格尔巴特的研究处在重要阶段的时候，他的研究中心失去了资助，很多人都离开了，最后又是剩下固执的恩格尔巴特继续为了自己已经奋斗几十年的梦想而努力。”

到了 20 世纪末，恩格尔巴特设计的具有友好用户界面、多窗口显示器、鼠标等人机交互能力的计算机系统终于获得人们的认可并迅速发展起来。这时，人们才意识到原来几十年前这位杰出的思想家和实干家就已经在默默地为之而奋斗。

尽管如此，恩格尔巴特并没有获得与他的贡献相当的名誉和财富，他仍然继续追求着很早之前就为自己立下的目标，或许这样他才能更加快乐。

有句古语说得好：“小人恒立志，君子立恒志”。总结一下，在为自己定下目标的时候遵循的几个基本原则如下：

- 量身打造，适合自己。
- 不能白立，切实真干。
- 不在宏伟，贵在不变。

3.2.3 让自己知道今天该干什么

如果将每天的成果比喻成砖瓦的话，有目标和无目标的奋斗就犹如长城和砖垛一样。心中有目标，每一步的实现必然是向着成功越来越远，最终将一块块砖瓦垒成了万里长城；而没有目标的努力，虽然也会生产出漂亮的砖瓦，但却仅仅是整齐地摆放在一起，永远都还是只是原材料。

“师兄，即便是定下了目标，做了长远的规划，我还是对于每一天该做的事情很茫然。光看着目标在前，但却追不上去。”

“有你这种想法的人也不在少数。目标有了，但具体到当前这一天、当前这一小时该干什么，就有些犯难了。”

“那师兄你帮我解答一下呗。”

“别急，我还是再来给你讲个故事吧！”

没有目标的时候迷茫，有了目标后却彷徨。很多立下目标的人可能都会有这种感受，苦于不

知道该如何继续自己的奋斗。读完下面这则类似喜剧的小故事，或许能给像蔡佳娃这样茫然不知所措的人一些指导性的意见。

一位女士（姑且叫做 Mary）刚刚过完自己的 30 岁生日，但她并不开心，于是，她找到了自己的好友吐露自己的烦恼。Mary 告诉自己的好友，现在的状况完全不是自己很早之前规划好的。好友就问她原先的规划是什么，Mary 说她的想法是在 32 或 33 岁的时候已经做妈妈了，但是她现在非但没有结婚，而且正在交往的男朋友比她年龄还要小，短期内根本没有结婚的打算。

好友说：“那你是不是很想坚持你自己的计划呢？”

Mary 说：“是的，我就是为这个而沮丧的。”

好友说：“那我们来分析一下。你想想看，我们假设你要在 33 岁的时候做妈妈，那么你应该至少在多少岁的时候结婚呢？”

“至少 32 岁，最好 31 岁。” Mary 回答说。

好友接着问：“那我们假设你在 32 岁结婚，那你应该在多少岁的时候认识你现在的丈夫呢？”

“至少两年前，也就是 30 岁的时候。” Mary 说。

“那你现在遇到了没有啊？”好友继续问道。

Mary 想了想自己现在的男友，无奈地摇摇头。

好友说道：“分析到这，你应该明白现在该干什么了吧？”

Mary 想了一会，眼前一亮说道：“恩，我现在应该和现在的男友分手，努力让自己去结识一个打算结婚的男士去！”

故事中 Mary 女士的目标就是最晚 33 岁的时候当上母亲，而 Mary 所做的却不是朝着目标在前进，甚至是在慢慢偏离自己的目标。Mary 的朋友帮助她巧妙地分析了一番，在时间上从将来往现在推导，由远及近回到现在，Mary 才发现自己当下应该处于什么阶段。

故事虽然有些搞怪，但是其中的道理却值得提炼出来深思。不像一般人从现在开始幻想到将来，我们可以从将来的某一天往现在回溯，这样自己职业生涯中每个阶段的状态都清晰明朗起来。将长远目标彻底分解成一个个的阶段状态，这样就不会再感到迷茫了。

Mary 的故事是从时间的角度上来考虑的，下面这个例子提供了另外一种规划的方法。

“蔡佳娃，怎么样？从这个故事中得出什么启发了吗？”

“呃，故事倒是听明白了，只不过还没想清楚有啥含义。”

“哈哈，这其实是一个喜剧里面的笑话，不过其中的道理却是很值得我们学习啊！”

“那到底是什么道理呢？”

“我再给你出道数学题，不等式的证明：

求证 $\sqrt{3} + \sqrt{6} < 2\sqrt{5}$ ”

“啊？师兄你猛地一提我还真不知道怎么说。”

“还是我来说吧，解答如下：

因为 $\sqrt{3} + \sqrt{6}$ 和 $2\sqrt{5}$ 都是正数，

要证 $\sqrt{3} + \sqrt{6} < 2\sqrt{5}$ ，只需证 $(\sqrt{3} + \sqrt{6})^2 < (2\sqrt{5})^2$ ，只需证 $9 + 6\sqrt{2} < 20$ ，

只需证 $6\sqrt{2} < 11$ 即 $\sqrt{72} < \sqrt{121}$ 。所以得证。”

“怎么样？看出点什么了吗？”

仔细分析解答这个证明题的思路：要想证明一个结果，先找出最终结果的充分条件，找到后把它当作一个新的需证结果，再寻找其充分条件，如此循环，直至最后的需证结果是一个无须充分条件证明的公理，或是可以通过已知求得的定理。这样，问题就解决了。

实现目标有的时候就像解答这样的证明题一样，定下了一个很明确的目标，但是现在却不知道从哪入手来实现目标，那么就可以从实现目标的那天进行回溯，推导出每件事的因果关系，最后就明白自己当前该干什么了！

“师兄，我好像看出点门道了，这个数学题和前面的故事都是从结果往前推的是吧？”

“正解，所以我们在规划目标的时候如果对于当前该干什么很是茫然，就可以用这种方法来让自己更清楚一些。”

“哦，师兄，我开始明白你的意思了。”

“恩，非常正确。这下你应该不会再感到彷徨了吧？”

两则故事所阐述的道理各有侧重，第一个故事主要将未来到现在的时间划分成不同的阶段，而第二个故事注重于实现目标的各个步骤间的因果关系。但是两个故事都是按照从终点到起点的顺序对目标进行分析的，这才是最重要的一点。

对于那些有目标却很彷徨的人来说，故事中介绍的由未来及现在、由果及因的思考方式很值得借鉴，比如一个从事 Java 开发的人可以为自己的职业生涯做一个规划表，如表 3-1 所示。

表 3-1 职业生涯规划表

工作年限	需达到的目标	所需要求
5 年	项目经理	需要对此行业领域的业务流程十分熟悉，项目架构能力很强
4 年	年薪 8 万	在公司需要有一定的不可或缺性
3 年	Team Leader	某个领域需要有很丰富的经验，需要技术能力水平很高
1 年	站稳脚跟	掌握流行技术，能适应开发这个行业
学校	顺利毕业，准备求职	需要通过学校考试，通过 SCJP 认证，需要有项目经验

细心的读者会发现，这个表是从上往下填写的。如此一来，就不会再为了自己清晰的目标感慨迷茫的现在了。在所要达到的目标旁边是达到该目标的要求，这样面对当下就不会无事可做。不过需要记住的是不要让目标只存在于确定的瞬间，定下长期目标并将其分为各个阶段，持续为之努力，收获的将是与汗水等价的成功与喜悦。

3.3 IT 认证的问题

曾经有个笑话说现在的社会已经进入了“纸器时代”，即人们所接触和追求的东西都是纸做的，如户口本、金钱、房产证、学位证等。在校大学生在为自己备战求职的时候，也免不了去寻找厉害的“纸器”来武装自己。证书，应该是大学生最能拿出手的纸制武器了。

3.3.1 认证那点事

通过学校教育获得的毕业证书、学位证书，应该是最常见的一种证书。对于 IT 专业的学生来

说，除了这种学校的文凭证书，现在的 IT 行业还有种类繁多的各种认证，国内的如计算机等级考试、软件工程师等，国外的如思科的 CCNA、Sun 的 SCJP 等。

“师兄啊，我们现在有些同学都在忙着考证，我是不是也该考个证壮壮胆啊？”

“你的同学们都准备考什么证啊？”

“很多种呢，有的在准备国家的软件考试，有的在准备思科的 CCNA，还有的打算考 Sun 的 SCJP……这么多的认证，搞得我也很茫然。这个铺天盖地的证书到底在我们求职的路上扮演什么样的角色呢？是不是必须拥有的呢？”

“呵呵，你的困惑不无道理，证书目前在 IT 职场上确实有着举足轻重的作用，单看当下如火如荼的各种认证培训就可以明白。那么在回答你的问题之前，我先给你讲讲认证是如何在中国的 IT 行业中诞生和发展的吧。”

1. 认证的作用

“首先，我们先来谈谈 IT 认证的作用。一句话，IT 认证是作为进入职场的敲门砖或通行证而存在的，在这一点上和学校颁发的文凭证书是没有太大区别的。”

“那也就是说 IT 认证的生命周期也就是求职那一场恶战呗。”

“一旦进入了 IT 职场之后，手中的认证就基本失去了作用，就像买完票进电影院后票就已经作废了一样。”

既然 IT 认证主要作用在求职这道关，那么其存在的价值之一就是弥补学校文凭在体现个人能力上的不足。而且因为目前很多不同领域的 IT 认证都是该领域的技术巨头公司组织开设的，所以一定程度上要比学校的文凭证书有更多的机会进入该领域的公司。

当然学校的文凭还是不可或缺的，至少它可以让你挺着胸脯说“我是大学生！”。实际也是如此，IT 公司在招贤纳士时首先要看是否为大学生，不是大学生就基本没有任何机会了。

2. 认证出现的原因

“根据刚才说的 IT 认证的作用不难推断出来，IT 认证出现的原因肯定是和用人单位的招贤纳士的方式和标准息息相关。国内的 IT 认证出现于 20 世纪 90 年代，由于改革开放后教育产业的发展，人才与职场的供求关系已经发生了很大的变化。”

“师兄，那之前是什么样的职场现状啊？”

“在那之前很多大学还是负责分配工作的，就算是需要自己找也只是写一份简历即可，因为很少会有人会被拒绝的。”

“看来做一个 90 年代的大学生很幸福啊！”

不过，好景不长，慢慢人才的竞争开始激烈起来，学历开始成为区别人才水平高低的分水岭。而随着发展的继续，单单靠学历已经不能够保证找到一份满意的工作了，能力开始作为用人单位衡量人才的最重要标准。

在用人单位对求职者能力重视程度不断加大的同时，相比较于学历，能力作为一个在当时很难判断的指标也开始越来越困扰着每个 IT 公司。而随后出现的 IT 认证则开始在一定程度上成为了求职者的能力水平的表征，此时认证的含金量是极高的。

3. 认证的诞生和发展

“师兄，既然 IT 认证出现的原因是用人单位对于求职者学历和能力的重视程度发生了变化，那么它是如何诞生的呢？”

“首先在中国竖起 IT 认证大旗的就是响当当的微软帝国，眼光独到的微软在 90 年代末第一个在中国开设微软技术认证 MCSE 的培训，从此就拉开了认证培训大发展的帷幕。”

“那微软认证的效果如何啊？”

“那时是相当火爆啊，那个时候报名参加培训的人特别多。当时的微软 MCSE 讲师非常辛苦，必须每天喝中药护理嗓子，因为每天从早到晚不间断地有培训工作。”

IT 认证从诞生到现在，一直作为在一定程度上代表能力的衡量标准而存在，很多时候用人单位对于认证的青睐甚至都超过了学历。因为 IT 认证考试中的评判标准都是由企业来制定的，其技术含量比起学校颁发的学历，更能和业内的行情保持同步。

在微软的带动下，很多公司认识到设立自己的 IT 认证对于招聘人才和公司发展的重要性，于是 Sun、Oracle、RedHat 等 IT 巨头们也纷纷推出自己的 IT 认证。而国内由国家的有关部门牵头，也设立了很多国家级的 IT 认证，这些在后面小节中都会为读者介绍。

3.3.2 现在的认证

中国市场上的 IT 认证发展到今天，认证种类和体系多种多样，含金量也高低不同，本节就来简单介绍一下行业中几种主要的认证。

“师兄啊，你刚刚给我讲了认证的作用和发展史，我都听明白了，接下来你还是讲讲我最关心的吧！现在 IT 行业主流的认证都有哪些啊？”

“呵呵，好吧，下面我就谈谈国内目前存在的国际认证和国内认证，让你有个比较深入的了解，这样如果你有打算的话就可以选一个比较厚重的敲门砖。”

1. 微软

微软打响了中国 IT 认证的第一枪，所以微软的 IT 认证是非常有价值的。在技术不断发展的同时，微软的 IT 认证也在不断地发展变化。现在微软的 IT 认证种类有很多，不再是最初的 MCSE 和 MCS D 那样的绝代双骄了，主要项目如下所示。

- MCSE (Microsoft Certified System Engineer)

微软认证的系统工程师，这是微软传统的认证项目，一个合格的 MCSE 能够在 Windows 平台下进行软件或服务的部署，各种服务器的配置和数据库的管理等。所以要想考过 MCSE 必须要掌握 Windows 的网络及操作系统知识和 SQL Server 的配置及管理，以及其他 Windows 网络服务器组件的管理。

- MCS D (Microsoft Certified Solution Developer)

微软认证的解决方案开发人员，通过这种认证的技术人员可以在 Windows 平台下进行主要是面向企业级的程序开发。这种认证侧重于程序开发，但是并不强调具体的编程语言。通过这种认证需要对项目解决方案方面的知识非常熟悉，同时也要适当了解桌面应用程序和企业级应用程序的开发模式。

- **MCDBA (Microsoft Certified Database Administrator)**

微软认证的数据库管理员，通过这种认证的技术人员，可以认为其对于主要是 SQL Server 系列的数据库有比较深的了解，并且能够熟练对其进行开发和管理。通过这种认证需要对微软平台下的 SQL Server 有很强的驾驭能力，而且由于 MCDBA 面向的领域较窄，只涵盖数据库知识，所以很多情况下 MCDBA 都用来和其他认证如 MCSA 等一起来考。

- **MCAD (Microsoft Certified Application Developer)**

微软认证的应用程序开发人员，这是达到 MCSA 之前的一个初级阶段，通过这种认证的技术人员，能够熟练地开发 .NET Framework 下的应用程序。要想通过这种认证，必须掌握微软平台下的 Web 开发技术，包括服务端和客户端。

- **MCDST (Microsoft Certified Desktop Support Technician)**

微软认证的桌面开发人员。这种认证是随着桌面操作系统的发展而产生的，同时也能够弥补微软在 PC 端认证的空白 (MCSE 和 MCSA 均是面向企业级的)，通过此认证考试的技术人员能够为 Windows XP 及以上的用户提供桌面系统的技术支持和系统故障的排除等服务。

考试费用：约 400 元/科。不同的认证由不同的科目组成。

纯金指数：★★★★☆

以上给出的只是微软几种比较常见的 IT 认证，微软的 IT 认证种类繁多，而且随着技术的变化也在不断地改变，每年都会有认证被取消不再使用，有的证书每年还必须进行再次审核发放。因此，建议有意向报考微软 IT 认证的读者在报考之前了解一下微软最新的认证政策。

2. Oracle

Oracle 是目前仅次于微软世界排名第二的软件公司，主要从事数据库软件的业务。其主要面向的是大型的企业级数据库应用，同时也兼顾企业级应用软件如 ERP 和 CRM 等的开发。Oracle 推出的认证有 OCP 和 OCA，OCA 一般作为 OCP 的一部分来考。

OCP (Oracle Certified Professional)，甲骨文认证专家，通过此认证的人，将具有对 Oracle 各版本数据库进行熟练管理和设计开发大型应用的能力。Oracle 在大型数据库软件市场上占有绝对的优势，所以 Oracle OCP 认证的含金量很高。

考试费用：10000 元左右

纯金指数：★★★★★

Oracle 认证的高含金量也体现在其高额的考试费用中，不同的版本考试费用是不一样的，Oracle 10g 的费用就比 Oracle 9i 要低廉许多。

3. RedHat

RedHat 认证是 Linux 操作系统中的权威认证之一，在 Windows 世界微软的认证是一家独大的，而在 Linux 世界存在的很多不同系统版本的认证中，RedHat 的 RHCE 认证具有明显的统治地位。很多其他的 Linux 版本如 TurboLinux 等都以 RedHat 的认证作为标准。

RHCE (Red Hat Certified Engineer，红帽认证工程师)，通过此认证将具有对 RedHat Linux 操作系统的安装、配置及管理以及故障排除的能力，有点类似于 Windows 平台下的 MCSE。

RHCE 考试分为三个部分：第一部分是上机操作进行故障排除（Debug Exam）；第二部分是多项选择题（Multiple Choice Exam），主要考 Linux 系统的各种命令；第三部分是服务器安装和网络服务配置测试（Server Install and Network Service Setup Exam），考生要根据要求对 Linux 系统进行指定服务的安装和网络配置的测试，并最终给出报告。

考试费用：每个部分的费用大约是 2500 元，如果一次不过第二次考就会变成 1000 元。

纯金指数：★★★★☆

Linux 操作系统目前正在快速地壮大，而作为 Linux 认证中的佼佼者，RHCE 应该是每个有志于从事 Linux 开发的技术人员的一张最好的能力证明书。

4. Cisco

作为国际上最大的互联网解决方案提供商之一，思科公司的认证在业界有很高的认可度，思科公司主要提供网络硬件如路由器交换机及网络软件如视频会议和防火墙等。思科公司的认证主要是面向路由器和交换机的配置与管理，按级别由低到高分为 CCNA、CCNP、CCIE。

- CCNA（Cisco Certified Network Associate）

思科认证的网络工程师，它是思科认证系列中最基础的一个。CCNA 的相关考点主要是网络的基础知识和思科低端路由器交换机的基本配置和管理。CCNA 也分很多方向，如 CCNA Voice（网络语言技术）、CCNA Wireless（无线技术）和 CCNA Security（网络安全技术）等。

- CCNP（Cisco Certified Network Professional）

思科认证的网络高级工程师，是思科认证系列中实用价值比较高的一种认证。作为思科的主要认证，CCNP 认证要求能够对网络节点在一定规模以上的局域网或广域网进行思科系列路由器和交换机的配置与使用。

- CCIE（Cisco Certified Internetwork Expert）

思科认证的互联网专家，是思科认证系列中最高级的一种，也是国际互联网技术中最有权威的认证之一。CCIE 认证分为路由交换认证和网络安全认证等几个类别，是思科系列中最难考的一种认证，因此其含金量也是最高的。

考试费用：CCNA 和 CCNP 的考试分为考试费和资料费，资料费大概在 1000 元左右，考试费中每一门的费用大约是 1500 元。CCIE 考试分为资格考试和实验考试，前者费用大概在 3000 元左右，后者在 13000 元左右。

纯金指数：★★★★☆

思科作为互联网市场的领军人物，其认证的含金量自然不在话下。在路由器交换机领域，思科的认证还是有相当强的说服力的。

5. Sun

Sun 公司的主要认证的就是针对 Java 技术与 Solaris 操作系统的，其中关于 Java 技术的认证保有量比较大。同时，由于 Sun 是 Java 技术的发明者，所以其在 Java 认证领域有着非常高的决定权。Sun 公司按照 Java 技术的层次高低和面向领域的不同将 Java 认证分为 SCJA、SCJP、SCJD、SCWCD、SCBCD 等八种认证，其相对关系如图 3-2 所示。

- SCJA（Sun Certified Java Associate）

Sun 认证的 Java 助理程序员,它比较简单,面向基础。主要考查的是对 Java 基本语法的掌握,所以含金量不是很高,考量不是很大。

- SCJP (Sun Certificated Java Programmer)

Sun 认证 Java 程序员,是 Java 认证里面最具有代表性的,同时也是人气最高的一种认证。SCJP 要求考生掌握 Java 技术的核心知识,并熟悉常用的 API 用法。SCJP 通常作为 Java 开发人员步入职场的一个敲门认证,有较高的含金量。

- SCJD (Sun Certified Java Developer)

Sun 认证 Java 开发人员,此认证需要首先通过 SCJP 认证才可以报考。SCJD 认证考试分为项目开发部分和项目论述部分,在第一个部分会被要求开发一个指定应用的设计方案及实现,然后在第二部分对其进行讲解和论述。

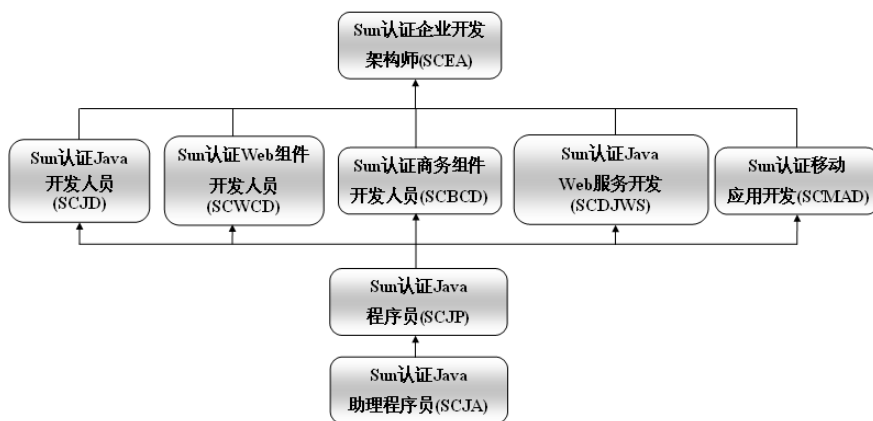


图 3-2 各种 Java 认证之间的关系

第一部分时间很宽裕,可以自己在学习工作之余慢慢完成,完成后提交即可。而第二部分则不同,是以面试答辩的方式进行的,而且要求使用英文,有一定难度。

- SCWCD (Sun Certified Web Component Developer)

Sun 认证的 Web 组件开发人员,此认证也需要首先通过 SCJP 认证才可以报考。此认证主要考查使用 Java 技术开发 Web 应用的能力,需要的知识涉及 JSP、Servlet、JavaBean 等 Web 开发相关的内容。

- SCBCD (Sun Certified Business Component Developer)

Sun 认证的商业组件开发人员,此认证同样需要首先通过 SCJP 认证才可以报考,主要考查的内容是开发部署 EJB 组件的相关知识。

- SCDJWS (Sun Certified Developer For Java Web Services)

Sun 认证的 Web 服务开发人员,此认证也需要首先通过 SCJP 认证才可以报考,此认证和 SCWCD 的区别就是对 Web 服务程序的设计有较高的要求。

- SCMAD (Sun Certified Mobile Application Developer)

Sun 认证的移动开发人员,此认证同样需要首先通过 SCJP 认证才可以报考。SCMAD 主要的考试范围是 Java 在移动设备如手机或嵌入式设备中的应用开发,主要用到的知识是 Java ME 领域的。

- SCEA (Sun Certified Enterprise Architect)

Sun 认证的企业级架构师, 此认证是 Sun 认证中级别最高的认证, 而且不要求通过前面的认证即可报考。SCEA 的考试分为三个阶段: 第一阶段进行架构基础知识的测验; 第二个阶段要求进行一个实际项目的架构设计; 第三个阶段需要针对第二个阶段设计的架构进行讲解和分析, 也就是面试答辩。

Sun 认证的考试全年都可以报考, 只要在考试之前到授权考试中心进行预约即可。与其他国际 IT 认证相同, Sun 的认证考试大部分都是英文版的, 因此对专业英语水平也有一定的要求。

考试费用: Sun 的认证考试每年都会有优惠期, 其他情况下大概都是 1500 元左右。

纯金指数: ★★★☆

6. 国内 IT 认证

国内的 IT 认证是在国际 IT 认证的迅速流行之下开始的, 起步较国际 IT 认证晚。国内的各种认证一般都是由劳动部、工信部(原信息产业部)、教育部来颁发的, 除了那些和国际厂商类似的表征能力的认证之外, 还有一些认可度比较高的如律师资格证、教师证等上岗证书。

国内的 IT 认证主要有计算机技术与软件专业技术资格考试, 软考是国家级的 IT 专业技术人员从业资格考试, 分为软件设计和网络管理两个大类别, 每个类别划分为不同的等级, 分为技术员、助理工程师、工程师、高级工程师。

国内的 IT 认证既是职业资格考试, 也是水平资格考试, 同时还具有职称性质, 而且国内的 IT 认证可以在日本、韩国与某些认证互相承认。

考试费用: 国内 IT 认证的考试费用根据不同等级收取不同的考试费, 一般来说为 100 元到 300 元左右, 同时因城市的不同而略有差异。

纯金指数: ★★★

国内的 IT 认证在国内有一定的肯定度, 但是总体和国际的 IT 认证比起来, 专业性要差一些, 类别的划分上也不够细致。



提示 本书中列举的各种认证仅供参考, 具体认证政策需以最新的官方消息为准。

3.3.3 该不该考个证

“师兄啊, 听你说了这么多 IT 的认证的内容, 那么对于我们这些一两年后就要杀进职场的人来说, 该不该搞一个证书呢? 搞个什么样的呢?”

“就现在来说, 证书并不能绝对代表一个人的能力, 因为现在的用人单位比以前聪明多了, 通过笔试面试这几关就能大致了解一个人能力水平, 而且很多情况下用人单位对于求职者的技能要求也不仅限于 IT 认证所包含的内容。”

需要说明的是, 尽管 IT 认证的作用已经不像前些年那样具有决定性, 对于即将踏上求职路的 IT 新手也并不是不可或缺的必备品。但是拥有一张漂亮的 IT 证书, 还是很能为自己的形象加分的。尤其是拥有一张高分的认证成绩单, 那将会让招聘单位刮目相看。

在某些情况下, IT 认证在求职的过程中所起的作用还是非常大的, 因为 IT 认证都是由企业带头设立的, 其技术特点和所依附的企业有很大关系。有时候同一类型不同公司的 IT 认证可能所

需求的知识是千差万别的，如 Cisco 的网络工程师可能就无法通过华为的网络工程师认证。

因此，遇到一些对技术要求专精的岗位，如数据库管理员、网络工程师等，最好还是有一证在手，因为那种 IT 认证对于个人能力水平的代表性很高。

“师兄，听了你这么一说，我想我还是以后考一个证书比较好，而且最好是拿到高分，这样的话或多或少也会为自己的技能增加一点说服力，是吧？”

“恩，说得没错。没有证书不代表吃不上饭，但是有证书或许能保证你早些吃到。”

“那师兄，我应该选择哪种 IT 认证比较好呢？国际的还是国内的呢？”

“我建议你还是选择国际的 IT 认证比较好，因为国际的 IT 认证比较难考，含金量也比较高，考下来比较有成就感。同时国际 IT 认证的考试费用也是相当高；而国内的 IT 认证就比较大众化，在专业程度和知识深度上都比国际 IT 认证要欠缺一些。”

总体来说，国外的 IT 认证发展比较早，其划分的体系也更加完善，所囊括的技术知识也比较先进全面。同时由于国外做 IT 认证的都是那些在相应领域一言九鼎的 IT 巨头，相比国内的来说要更加高端和专业，所以其认证也自然要显得雍容华贵一些了。

国内的 IT 认证则相对来说门槛低，更容易通过，在技术水平上没有国际 IT 认证的代表性高，这也使得其权威性要稍差一些。最重要的一点，国际的 IT 认证在很多地方都有效，认可度很高；而国内的 IT 认证则含金量不够高，只能在国内使用。

3.4 本章小结

IT 这个江湖令人向往也令人畏惧，想要从学校的深山中走出来，顺利地进入 IT 江湖，没有充分的准备是不行的。希望各位读者珍惜自己在山中的时间，为了心中的目标暗自努力，深学苦练，争取让自己的下山之路平坦顺利。



Java程序员 职场全攻略

——从小工到专家

吴亚峰 著

懵懂菜鸟如何才能成长为一名成功的开发人员？

如何在“血雨腥风”的求职大战中脱颖而出？

职场中只要技术过硬就可以吃遍天下吗？

是学招式重要，还是练内功重要？

掌握了哪些技能，才算是真正的高手？

.....

资深技术专家，娓娓道来。



 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

第 4 章 必须通关的游戏——求职之旅

如果把找工作这个必经之路比做打游戏的话，它与真正游戏的主要区别就是必须通关。否则不论打到第几关都算输，没有水平高低，只有成败。而这个游戏，除了极少的特殊情况，绝大多数的求职者都是按照投简历、笔试、面试、试用期这条路求职成功，最后转正成为正式员工。

就像游戏也会有 bug 和秘籍一样，任何竞赛性质的活动都是有对应技巧的。本章旨在向那些摩拳擦掌一腔热情，豪气冲云天的同学们介绍每个环节的通关攻略。同时，也会向那些一咬牙一跺脚气急败坏决心造假的读者们分析一下每一关卡的蒙混几率。不过这些所谓蒙混几率就像所有方便面包袋上的效果图一样，旁边必有一行小字：“仅供参考”。

就像前几章一样，我们憨厚勤奋的主人公蔡佳娃同学一直在师兄牛开复的指点下朝着一个 IT 牛人的目标稳步前进。现如今蔡佳娃已经升入大四下半学期，初出茅庐的他就要信心满满地开始求职之旅了。让我们追随他的脚步，看看现今的职场风云吧。



提示 在这里要提一下另一个求职区别于打游戏的特点，那就是虽然你可以失败了重新来过，但是切记，关卡已经变了，上次失败的经验下一次可能并不会有用。出于讲解的需要，我们的蔡佳娃在这本书中是一粒蒸不烂、煮不熟、捶不扁、炒不爆、响当当的铜豌豆，屡战屡败，屡败屡战。

4.1 简历靓起来

好的开始是成功的一半，只有简历被看中才会有资格进行笔试和面试，才算真正进入了游戏。同时这一关也是比较困难的，因为这一关玩家太多，数据的真实性最不可靠。而且就算一个求职者于情于理都应该通过，如果出现简历书写上的缺憾，那也只能仰天长叹，从头再来。

4.1.1 简历不是这样写的

蔡佳娃目前对于工作这件事还并不是很着急，只是准备提前感受一下这个过程。即便如此，在满怀信心地投出第一份简历后，却是泥牛入海，毫无消息。这还是让蔡佳娃有点小郁闷，下面请看蔡佳娃简历的第一版。

<h2>个人简历</h2>	
I 个人信息	
姓 名：	蔡佳娃
性 别：	男
出生年月：	xxxx年x月xx日
籍 贯：	xxx省xx市
	

毕业院校: xx理工大学
学 历: 大学本科
专 业: 计算机科学与技术
联系电话: xxxxxxxx E-mail: xxxxx@xxxxx

II 自我评价

待人真诚, 对待工作认真负责, 有团队意识, 能吃苦

VI 专业技能

学过数据结构、操作系统、计算机原理、计算机体系结构
学过 IBM-PC 汇编语言程序设计、C 语言程序设计、C++ 程序设计、Java 网络编程
学过数值分析、网络协议、电路、数字逻辑、离散数学、大学物理、计算机体系结构
学过高等数学, 大学英语, 熟悉 HTML/CSS/JavaScript

专家分析: 这份简历放到考官面前会被直接“秒杀”, 原因是什么呢?

- 专业技能中列出了所有在大学中学过的课程, 这就是被“秒杀”的主要原因。考官通过简历要查看的是应聘者的专业技能, 而不是应聘者所在学校的教学计划。
- 但凡对自己能力自信的学生, 都会优先地将自己的特长和精通的技能写到简历中, 根本不会为学校的课程留有大量的篇幅。

因此, 这种简历遭到舍弃也就无可非议了。

蔡佳娃是个好琢磨的人, 他略加研究, 觉得自己的专业技能过于小儿科, 不够时尚, 搜肠刮肚般回顾了一下自己在大学里学过的技术, 便又推出了第二版简历。

个人简历

I 个人信息

姓 名: 蔡佳娃
性 别: 男
出生年月: xxxx年x月xx日
籍 贯: xxx省xx市
毕业院校: xx理工大学
学 历: 大学本科
专 业: 计算机科学与技术
联系电话: xxxxxxxx E-mail: xxxxx@xxxxx



II 自我评价

待人真诚, 对待工作认真负责, 有团队意识, 能吃苦

VI 专业技能

精通 C/C++、VB、HTML、CSS、JavaScript

精通 .NET、C#，熟练使用 SQL Server

精通 Java SE 类库、Java EE、Java ME、JSP、Servlet

熟悉 Linux、Solaris 等操作系统

精通 Photoshop、Flash，熟悉 EDA，熟练掌握 C51 单片机

这个看似非常完美的简历应该没有什么纰漏了吧，蔡佳娃这样想着，就满怀信心地开始了第二次的简历大投送。在经过了好几天的寂静，又接了两个拒绝自己的电话之后，蔡佳娃不得不开始再次审视自己的“完美”简历。

专家分析：第二版简历看起来很漂亮，为什么会无人问津呢？

- 这份简历水分很大，估计很多考官会这么想。因为所有 IT 业界的“过来人”都明白，短短四年的大学时间，是不可能精通这么多专业技能的。应聘者明显是在夸大事实，不够诚实。
- 此类简历也暴露出应聘者对自己的不了解和不自信，学过这么多技能，居然不知道自己最擅长的是什么。或者觉得自己没有出色的地方，只好实行平均主义，“精通”了所有的技能。

于是乎，这种“水货”简历也难免遭到“秒杀”。

蔡佳娃犹豫了一下，决定去掉简历上一些出镜率太高的“精通”，于是“精通 Photoshop”、“精通 .NET”、“精通 Flash”等被删掉了。

这么一来，蔡佳娃心里的惭愧感才少了一些。毕竟他只用 Photoshop 将图片的背景处理成透明，Flash 也只是搞出过地球围绕太阳转的引导动画。而至于 .NET、VB 等，如果看过书就叫做精通的话，那对于自己真正学过、研究过、思考过的 Java 他应该用哪个词来修饰呢？

如此一来，蔡佳娃便一不做二不休，大方地删掉了“熟悉 EDA”、“精通 C51 单片机”……

这么一来，蔡佳娃的简历干净了许多。专业技能里面只剩下“精通 Java 类库”、“精通 Java ME”、“精通 JSP、Servlet、HTML、CSS、JavaScript”等与 Java 有关的内容。浏览一遍，蔡佳娃又在个人评价中加上了一句“热爱学习”，然后直奔复印店。

不用说，第三版简历仍然还是泥牛入海，没有消息。

专家分析：虽然蔡佳娃一直在改进，但是第三版简历还是有欠缺的地方。

- 第三版简历乍一看没有什么问题，可是专业技能还是不够具体，比如 Java EE 这方面如果加上“熟悉 WebWork2、Spring、iBatis、Lucene 等开源框架的使用”和“熟练使用 SQL，熟悉 Oracle、DB2 等大型数据库下的 SQL 开发”等更加具体的内容，考官便会对应聘者比较丰富的知识留下一个好的印象。
- 另外，蔡佳娃最后加上的一句“热爱学习”也会为他减分不少，因为用人单位更愿意招聘一个来到公司就能立刻上前线为公司创造财富的成手，而不是一个需要先消耗单位资源学习一通才能胜任工作的员工。

其实，蔡佳娃的三份简历一直得不到用人单位的肯定，很重要的一个原因就是没有主旨。简历和文学创作一样，需要一个主旨，然后所有自己添加上的知识、技能和经验都为一个主旨服务。这个问题在写简历的时候就应该弄明白，而不要等着考官来研究你简历的中心思想。

4.1.2 写出出色的简历

上一小节我们初识了简历关的困难，在“欣赏”了一下蔡佳娃同学的简历之后，也分析了其失败的原因。写出一份让人眼前一亮的简历对于打响冲锋职场的第一枪十分有必要。请看蔡佳娃是如何在高人的指点下重返战场，勇闯简历关的。

简历上的失败让蔡佳娃很是郁闷，不得不求助于刚出差回来的师兄牛开复。

困惑的蔡佳娃带着自己简历的三个版本和最近尚未问世的第四版找到了师兄牛开复。牛开复看过简历，把蔡佳娃狠批一顿，接着便向毫无头绪的蔡佳娃讲了一些自己的求职心得。

“蔡佳娃，我当时找工作从来都不是一份简历投到死，每次投简历之前都要先了解一下用人单位的岗位需求，尽量让自己的简历向那个方向靠拢。这样全神贯注地射出一箭，比起漫无目的地乱开枪命中率要大得多。”

“师兄，我们宿舍有的人简历做得很精美，还放进去几张艺术照片……”

简历越是豪华，说明里面真正有价值的东西越少，不然不会花大手笔做表面文章。往往这种简历用人单位是很嗤之以鼻的。写这种简历的人，估计对 IT 这个行业了解太少，IT 开发行业是研究技术的，要求的是踏实的态度和严谨的作风。这种华而不实的简历不仅让人看着反感，也不实用。”

需要说明的是，作为拉动全球互联网技术前进的发动机，IT 行业在招聘人才方面越来越多地通过网上来进行，尤其是简历关，很多简历都是在网上传送的电子版。所以各位求职者在备战简历的时候也要为自己多开辟一个网上战场。

“师兄，那写简历的主要原则是什么呢？”

“简历不在字数多，篇幅大的简历反而让考官觉得很烦琐，同时也会淹没你的亮点。所以写简历就是要简单明了，旨在向用人单位展示你对于所招聘岗位的合适程度。”

“那简历的主要部分就是专业技能了？”

“也不全对，你的这几份简历有个共同的毛病，就是都没有写自己的项目经验，就像我之前对你说的，项目经验也是用人单位衡量一个人能力水平的标准，考官是综合你的专业特长和项目经验来判断你的能力是否能够满足他们的岗位需求的。”

“哦，那我回去把我之前做的一些项目加上去，我的项目经验不是很多，可不可以编造一些加上去啊？”

对于大学生来说，项目经验是个让人又爱又恨的东西。现实的情况是：大学生不仅要有一定的项目经验，而且必须是真实的项目经验，厉害的考官一眼就可以看出来简历上的项目经验是真是假，是大是小，是实是虚。套句古语就是：成也经验，败也经验。

就算在简历关能够用不真实的项目经验侥幸过关，但简历之后的笔试、面试，以及随后的试用期，都是在考查你这个人有没有撒谎，是不是像简历上说的那样有能力。

虽然很多人在写简历的时候都会稍微地把自己夸大一些，这或许已经是个心照不宣的潜规则，因为大家都夸大一些，如果你不夸大，你就是落后了，但是要注意这个度，否则，或者会被发现欺瞒而被淘汰，或者成功过关但到最后难免原形毕露。

“师兄，那我还是老老实实说真话吧。其他部分呢？比如自我评价、兴趣爱好？”

“这些东西用人单位并不是特别的关心，因为你的专业技能，项目经验都可以通过笔试、面试或试用期进行检验，而你的性格、爱好就不是那么容易试出来。况且相对于专业技能和项目经验，你的性格爱好对于为公司创造财富并不是最重要的，基本上是属于比较虚的东西。”

“呃，师兄啊，听了你这一席话，我倒是明白了不少。可是，对简历的整体把握上还是有些理不清的地方。”

“这哪是理不清啊，你就是懒得重新造，想有一个模板依葫芦画瓢是吧？”

“嘿嘿，还是师兄了解我啊。”

“好吧好吧，我把我以前找工作的一份简历给你参考参考，你大概按照这个样式和层次来制作你的简历，记住我说的简单明了就好。”

“谢谢师兄，师兄我对你的敬仰犹如滔滔江水，连绵不绝，又犹如黄河泛滥，一发不可收拾……”

“行了行了，别贫了。这回你要再不成功，你就别回来见我了。”

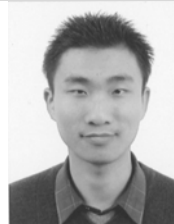
蔡娃娃拿到牛开复师兄的简历，的确是有种豁然开朗的感觉：

个人简历

I 个人信息

姓 名： 牛开复
性 别： 男
出生年月： xxxx年x月xx日
籍 贯： xxx省xx市
毕业院校： xx理工大学
学 历： 大学本科
专 业： 计算机科学与技术
联系电话： xxxxxxxx

E-mail: xxxxx@xxx.xx



II 自我评价

专业技能扎实、吃苦耐劳、富有钻研精神、团队意识良好。
擅长 J2EE 开发，熟练掌握 Lucene、Webwork2、Ibatis、Spring 等开源产品，
熟练掌握 Web Service、Socket、Http 等通信接口的应用。

III 项目经验

2006.10 – 2007.06 信息产业部通信软件工程中心项目的开发

项目描述：移动基站软件项目

项目职责：在研发部担任软件工程师，负责用 Java 开发移动基站软件

2007.08 – 2007.09 电子商务网站积分、手机话费支付接口

项目描述：E 动商网与湖北移动的电子商务网站积分与小额话费的接口开发，该接口使用 Web Service 与 Socket，主要供网站的展现层查询和业务调用。

项目中职责：

主要负责接口的设计和开发，包括结算算法的设计和实现、部分业务逻辑的封装等。

IV 获得奖项

2006 年 10 月 – 2006 年 12 月 参加由 Sun 公司举办的第二届 Java Cup 全国大学生信息技术大奖赛（NetBeans Plugins Code Jam），获得大赛三等奖。

V 认证情况

大学英语六级 613 分

2007 年获得了 SCJP 证书 成绩 93 分

VI 专业技能

- （1）熟练使用 Java SE 核心 API 及 JavaScript 语言。
- （2）J2EE 服务器：精通 Weblogic、JBoss 中项目的部署与运行。
- （3）持久层：精通 JPA、Hibernate 的开发与调试，有实际项目工作经验。
- （4）Web 层：精通 JSP、Servlet、JavaBean、Struts、JSF 的开发与使用，有实际项目工作经验。
- （5）业务层：精通会话 Bean 的开发与部署，对 Spring 也有较深的研究。
- （6）数据库：精通 Oracle 数据库的使用、脚本的开发、数据库性能优化、分布式环境的搭建。另外精通 MySQL 的使用，有实际项目工作经验。
- （7）掌握 Linux 系统常用指令，有在 Linux 下部署、维护 Java EE 项目的经验。

VII 兴趣爱好

篮球、足球、电脑游戏

蔡佳娃如获至宝，参考着师兄牛开复的简历，冷静地研究了一下用人单位的岗位要求，写出了第五版简历。然后，蔡佳娃将简历发给了自己之前分析过的几家比较合适的单位。

蔡佳娃并没有傻傻地等待回复，因为牛开复师兄告诉他，他更应该关心的是简历通过之后的第二关——笔试的问题。所以蔡佳娃一直在宿舍闭关研究 Java 技术。

过了两天，蔡佳娃给师兄牛开复打电话：“师兄啊，我收到了一个笔试通知！”

专家分析：总体上讲，蔡佳娃算是一个比较用功的同学，不应该在简历关就挂掉。只是还没有学会如何驾驭自己的简历让其博得考官的好印象。其实就像故事里牛开复师兄说的那样，好的简历大概遵循如下几个原则：

- 简明扼要

简明即指简历内容简单明了，相比较于洋洋洒洒篇幅很长的简历，考官更容易在精简朴素的简历中快速发现应聘者的亮点。扼要即指主题明确，能让考官在看到简历后就很容易根据应聘者的能力为其做一个定位。

- 匹配度高

简历信息与招聘单位岗位描述匹配度高，就像牛开复师兄说的一样，简历不应该通用一份。根据招聘单位的职位描述，应聘者也要适当调整，力求自己的简历所描述的技能与单位的岗位描述相匹配。

- 关键信息突出

关键信息突出，简历内容的顺序不需要依照传统，可以把需求相关度高的内容放到上面，比如在应聘一家对 Java SE 要求比较高的单位时，可以把自己 SCJP 的高分成绩放到个人信息的下面，然后再继续介绍自己的技能特长等。

4.1.3 如果是机器筛选简历

IT 行业在不断地改变着每个人的生活，最主要的表现就是把人从烦琐的劳动中解脱出来，或者至少是大大提高其效率。前面已经说到很多时候简历都是在网上交流，如今不仅如此，简历的初步筛选也开始渐渐由人力劳动转为由任劳任怨的机器搞定了。

“师兄啊，我们宿舍有个人，他的简历和我之前写的一份挺像的，在专业技能一栏里写了好多精通的技能。你不是说这种简历会被考官直接‘秒杀’吗？可是他和我一样都接到了笔试通知呢！”

“有这个可能啊，现在有些单位招聘人的时候是用机器来筛选简历的，机器虽然快，但机器是死的，永远也赶不上人脑聪明，只是按照固定的算法挑选出符合条件的简历。”

“那岂不是会有许多冤假错案了？”

“冤假错案倒是不大可能会有，更多的应该是漏网之鱼吧。因为机器筛选简历是根据简历中的内容与输入的招聘岗位需求关键字的匹配程度来进行的。所以只要把自己的专业特长写得全面一些，就可以骗过机器的眼睛了。”

“这种方法不太科学吧，大家如果都写完完美简历，机器会让所有人都通过的。”

“非也，机器一般只是起到初步过滤的作用，用人单位会继续人力审查机器挑选出来的简历，如果发现钻空子的‘完美简历’，依然会直接‘秒杀’掉。你们宿舍的同学应该还是比较走运，不仅逃过了机器的眼睛，还骗过了考官，不过到了笔试应该就不会那么走运了。”

应聘求职者越来越多，简历也像洪水猛兽般涌向了用人单位，善于创新的 Google 在几年前美国总部招聘中就率先采用了机器过滤简历来筛选求职者。随着数量越来越多简历铺天盖地飞向各个用人单位，机器过滤简历便逐渐成为各用人单位挑选人才的第一个大筛子。

但是考虑到筛选算法不可能完善到与人力筛选有相同的准确性，只是比较关键字的匹配程度，如果应聘者各方面能力都不错，但是表达方式有问题；或是应聘者能力并不是很出色，但是善于雕琢文字，都难免会发生故事里的“冤假错案”。

机器开始过滤简历或许并不是坏事，至少这个一丝不苟的考官会让应聘者不敢对自己的简历再有所怠慢。其实不论是面向机器还是人，在书写简历的时候都应该注意让自己的经验介绍和专业技能规范、清晰、醒目。

4.1.4 简历小结

通过前面几个小节介绍，相信某些读者可能会对简历这一关有所“想法”，会对于其公正性保守地表示怀疑，其实这一关的蒙混几率应该是很低的。根据笔者的经验，靠真功夫写出来的简历与迎合拼凑出来的简历还是有很大区别的。

这一关不仅会淘汰一些能力稍逊的应聘者，有时也会向有真才实学的应聘者宣布“Game

Over”。因为简历也是一个人表达能力、思考能力的体现。就算你胸中再有波澜，双手却无从掌墨，仍然是无法向考官要来第二关的入场券。



提示 小小一份简历，是每个应聘者走向自己职业生涯的敲门砖。简历即是自己的门面，对自己做一个透彻的分析，定位自己的职业目标，从而写出一份简明扼要、关键信息明确的简历，为自己的职业生涯走向成功开个好头，不要让简历的内容和形式亏了自己的才能。

4.2 笔试，混可不行

通过了简历关的“海选”，笔试这一关算是比较回归校园了。从起源于隋朝的科举，到现在的学校教育，考试一直作为一种相对简单而又比较公正的选拔人才的方式。本节将着重介绍笔试的考试范围和如何准备笔试。同时，在上一节如愿通过简历关的蔡佳娃同学也要开始他的第二关历险了。

4.2.1 初识笔试

如果说简历关是时好时坏、发挥不稳定的主观题，那么笔试关就是真切实在、考你没商量的客观题了。关于考试，每位同学的临场经验都很丰富，不再需要有人指点。而笔试和普通考试除了在内容上存在不同之外，考场纪律也不会有校园考试那么严明。

到了笔试这步田地，还想要作弊可是需要很大勇气的，不仅是要求胆量，还要有勇气面对将来有可能被戳穿的局面。同时笔试中面对着满屋子都是来抢这个饭碗的竞争者们的虎视眈眈，作弊的手段也非常有限了。

蔡佳娃也深知笔试马虎不得，所以简历发出去后便一直闷在宿舍准备可能到来的笔试。结果刚闭关了两天，发现 Java 实在是太博大精深了，于是又求教于牛开复师兄。

“师兄，我前两天投出去了几份简历，有了你的指点，果然接到了一家公司的笔试通知，就开始研究笔试了，师兄你再开导开导我呗。”

“笔试是个实在的东西，没什么技巧，就看你平时的用功程度和知识积累了。”

“那笔试一般都考什么呢？Java 这块蛋糕这么大，我怕我没时间和精力彻底吃掉啊。”

“看看，又想偷懒了。好吧，我大概介绍一下笔试的内容。鉴于你这种临时抱佛脚的情况呢，我建议你下大力度好好啃一啃核心 Java，比如 Java 中的内存管理、多线程开发、集合框架等，这都是一些比较基础但又非常重要的部分，高端的内容还是少研究比较好。”

“哦，这样就简单许多了呢。”

其实，很多读者都会有蔡佳娃这样的想法。觉得 Java SE 比较容易掌握，只是基础而已，而 Java EE 和 Java ME 相对来说更值得花费工夫去研究。其实基础并不一定简单，就像建大楼时地基是最费事也是最费资金的，只不过由于平时不能直接看见它普通人不重视罢了，承建商可是很重视的啊。

另外，大部分人都错在把 Java 仅仅当作一门语言来学习，而不是在学习不断体会面向对象的思想以及 Java 本身的平台特性，读者朋友们在这些方面要多注意下工夫。

“简单？看来你对 Java 领悟不够高啊。我来考考你，面向对象的四大特性是什么？”

“这个简单，抽象、封装、继承、多态。”

“好，那什么是多态？”

“啊？？多态是……一个父类……派生出多个子类，然后子类调用……”

这是一个很多人都知道但是却无法说清楚的概念，多态的含义是不同的对象有相同的一般轮廓或形态，可以执行某个相同的操作，但是执行过程和细节却各不一样。比如所有的机动车都有刹车的功能，但是不同车的刹车方式和效果却截然不同。

很多 Java 初学者就跟蔡佳娃一样，研究问题只局限在表面，学习 Java 只注重语法格式，只把目标限定在熟练掌握循环嵌套、分支和判断等流程控制这些超基础的内容之上，结果遇到像多态性定义等类似的问题就彻底瘫痪了。

“哦，果然其中奥妙多多啊！”

“再问你一个，Overload 和 Override 的区别是什么？”

“一个是重载，一个是重写。重写是子类对于与父类同名的方法有不同的实现，重载和重写差不多，是名字相同但参数个数和类型不同的一组方法，重载也属于多态性吧？”

“不完全对，方法重载是可以提供类似多态性的好处，但是它与多态机制有本质的区别，重载一般指在一个类内部的方法重载，重载不同于覆盖，重载的方法是可以改变返回值类型的。最重要的是，重载不是面向对象专有的。”

“听君一席话，省我十本书啊！”

“说到书，我可以向你推荐一本讲核心 Java 的好书《Java SE 6.0 编程指南》，这本书讲得很全也很透彻，你可以好好读一读。”

“听说过，哪天拜读一下。这样按师兄你说的准备应该差不多了吧？”



提示 其实如果有读者心里太没底，可以在求职前去考一下本书在第 3 章提到过的 SCJP 认证，如果通过并且成绩在 85 分以上的话，笔试这一关中核心 Java 这部分过起来还是比较游刃有余的。但是，如果像蔡佳娃那样认为笔试只有这些，那就大错特错了。

“蔡佳娃，你就是有些毛躁，我话还没说完呢。笔试可不是只考你核心 Java 就了事的，不然大家都扬着 SCJP 的高分成绩单招摇过市了。根据用人单位的岗位需求，EE 和 ME 方面的知识还是要考的，但大多是一些描述性质的问题。”

“师兄啊，对于我们菜鸟，最好的说明手法是举例子，呵呵。”

“真服了你了，好吧。比方说问你 Java EE 是什么？”

“啊？？还真让师兄你说准了，往往常挂在嘴边的并不是我们彻头彻尾明白的，这个问题我还真没法回答呢。”

“所以说学习的时候钻得深是好事，但也总要不时的浮出水面把握一下全局。Java EE 是原 Sun 公司（现在被 Oracle 收购了）提出的多层（multi-tiered）、分布式（distributed）、基于组件（component-based）的企业级应用模型（enterprise application model）。”

“真是惭愧啊，这么久了才知道 Java EE 的身份唉。”

“像这样的题还有很多，比如解释 JNDI、JMS、SOAP、JTA、RMI、JDBC、EJB 技术，或者浅谈一下 Struts，描述一下 Servlet 的生命周期等。”

很多概念性问题都是这样，往往嘴上念得最顺溜的并不是我们最明白的，学习 Java 要不断地去思考和领悟，要通过 Java 这门语言本身感受到面向对象设计的思想。这是个自然而然潜移默化的过程，既不能过分强求于自身，也不要妄求转载自他人。

学习的结果全在自己的态度，如果把 Java 仅当作一门技术来学，那么 Java 也只能让你做一个手艺熟练的敲代码人员。但如果把 Java 当作一种面向对象编程的思想模式，那么 Java 也必将带领你进入一个更为广阔的程序开发天堂。

“哎，那师兄你说的这些知识点看看书应该好拿下吧！哈哈。”

“非也非也，有些东西是你看看就会记住的，有些东西却必须在有一定运用经验和领悟的基础上才能牢记和掌握的。近期可能来不及了，你以后一定要注意总结思考啊。”

“恩，师兄说得也太有道理了！”

“哦，对了。还有个部分笔试也会考到，那就是数据库方面的知识，因为但凡一些实用的有一定规模的应用程序，没有不用到数据库的。但是这方面涉及较少，而且这方面知识的积累更多是来源于实践。临时挑灯夜战估计作用也不会很大，但是参考一些常用的笔试题还是很有裨益的。”

“好的，现在我的思路开始有些清晰了。师兄，我会好好准备的！”

既然是考试，肯定要有统考大纲，本节大概讨论了笔试的考试范围，即核心 Java、Java EE 或 Java ME 方面描述性的原理或知识、数据库方面的知识。这三个部分所占比重一般依次递减。有了统考大纲，备战时就大致有了方向，不过有些东西像核心 Java 可以更多依赖读书和体会，有些东西却是在每一次的编程实践中领悟到的。



提示 请牢记我们为灵活运用 Java 而战，并非为熟悉 Java 而战，我们是实践主义者。前辈的经验告诉我们，只有理论联系实践才能真正开发出有价值的软件。

4.2.2 牛刀初试

一个礼拜过去了，蔡佳娃同学果然还在专心备考。牛开复师兄过来看他了。

“小蔡啊，研究得怎么样了？”

“大后天就要去笔试了，所以我现在临阵磨枪，不快也光啊。”

“呵呵，光看书也不行，来，我给你找来了一些笔试考题，让你提前练练兵。”

“不是吧，我怕我还没准备好……”

“哪有什么准备好不好的啊！做做看先！”

像蔡佳娃这样的人就是有些过于谨慎了，很多时候，事情都不可能等到准备得万无一失才去做。比如排练节目，每次排练都会觉得有瑕疵，不完美，但等到硬着头皮上台演出的时候，舞台上是什么样的，在观众眼中，那就是最终版本，是经过准备后的最完美展现。什么时候亮剑出鞘，什么时候就算准备好了。自信，有时候也是一种胆量。

“那师兄你考考我吧，看看我能不能一剑封喉呢？”


“小子你还挺不谦虚，好，先给你介绍一下笔试的题型吧。笔试大概有选择题、简答题和编程题。有的时候还会有一些智力题，或者叫谜语。”

“哇，内容好丰富，还有编程题和智力题。”

“是啊，用人单位为了招聘优秀的员工可是想尽了办法呢。”

“那我们就开始吧！”

蔡佳娃接过师兄的考题，埋头做了起来。

 **提示** 在准备笔试的过程中，可以适当地做一做笔试的经典考题。因为把书看懂了是一回事，能够在遇到问题时运用看懂的知识又是另外一回事。但是不建议大家把题目背下来，搞题海战术。因为抛去极小的成功几率不谈，本书倡导的可是一条光明正大的高手之路。

师兄看了看时间，说道：“差不多了，来，甭管做成什么样子，我们一起看看吧！”

蔡佳娃有些不甘心地吧试卷交给牛开复师兄，师兄略略浏览，说道：“恩，整体上还可以，不过有些地方就有些不够完美了，比如这道题：”&和&&的区别是什么？

“&是位运算符，表示按位与运算；&&是逻辑运算符，表示逻辑与。没错啊！”

“再想想，仅有这些吗？”

“不是吗？……哦，对了，&也可以作为逻辑运算符，表示逻辑与。”

“其他的呢？&在表示逻辑运算与的时候，和&&是一样的吗？”

“哦！&&是短路运算符，而&不是！”

“所以嘛，想问题一定要周全。Java 笔试中会有很多像这样的题目，要求你区别性质相似容易混淆或者仅仅名称相似含义相差甚远的两个概念。类似的题目比如：


- 区别 Collection 和 Collections
- final、finally、finalize 的区别
- sleep() 和 wait() 有什么区别？”

“第一个我知道，Collection 是集合框架类的上层接口，Set 和 List 均继承自它；而 Collections 类是针对集合框架类的一个工具类，它提供一系列的静态方法对各种集合框架进行搜索、排序、线程安全化等操作。”

“恩，不错。光说得流利是不够的。运用的时候别忘记了。剩下的那几道呢？”

“剩下的这几道题用的时候能分辨出来，但是要说，还真有些无语呢。”

“还是领悟得不够吧，没关系，冰冻三尺非一日之寒，慢慢练习慢慢体会就行。”

 **提示** 笔试中的选择题和简答题大概就是考查这些方面的内容，也还会有 Java EE 或 Java ME 以及数据库方面的知识。对于选择题还好处理些，主要是简答题对于应聘者的思维能力和表述能力有比较高的要求，读者应该在实践中多体会。

牛开复师兄继续说：“下面我们来看看你的编程题做得怎么样。”

题目：已知一字符串，长度不定，要求找出在字符串中出现次数最多的字符，并输出其出现次数。

“师兄，这个题目我的做法是先把字符串转换成数组，然后遍历这个数组，对于每一个字符，分别向前和向后查找有无相同的字符，然后记录下出现的次数，并与已有的最大值 max 比较，若比已有最大值 max 大则将此次出现次数赋值给最大值 max。”

“恩，这么想没错，但是你觉得考虑欠妥吗？如果某两个字符的出现次数相同且都为最大值呢？岂不是会漏掉？”

“对对，这个没有想周全。那我应该在后台维护一个整形数组，长度和字符串一样长。然后，与一开始的方法一样遍历数组，整形数组记录相应索引的字符出现的次数，同时得到最大值 `max`。然后再次遍历字符数组，只要它的出现次数等于最大值 `max`，就将其输出。”

“不错，这道题这么答可以。这算是大众解法。你再想一想还有什么方法吗？”

“别的嘛，我想想……想不到了，还请师兄指点一下啊！”

“我说的这个方法不算是最佳解决方案，但是却提供了一种新思路。那就是哈希表的方法。”

“哈希表？这里也可以用哈希表？”

“没错，我们把字符串生成的字符数组中每个元素当作要存放的值。新建一个字符数组为哈希表，长度为字符串的长度，这样能保证存下整个字符串。随便定义一个哈希函数，比如 $h(x)=x\%length$ ， x 为字符的 Unicode 值， $length$ 为哈希表的长度。”

“哦，我有点理解师兄你的意思了。然后是不是进行存储？把字符数组中的每一个字符都通过哈希函数计算出地址，如果地址冲突，就判断待存字符是否与已存在字符相同，不相同则根据处理冲突方法继续寻找散列地址，相同则计数器加 1。”

“恩，最后和第一种方法一样，先遍历出最大值 `max` 再把所有与最大值 `max` 相等的字符输出即可。怎么样，这种方法是不是让你眼前一亮啊？”

“的确是这样，能想到这种方法，肯定对算法有很深的体会。”

专家分析：编程题一般分为两种。

- 第一种主要考查应聘者对 Java 语言的驾驭能力，比如会考多线程编程、内部类的实现、继承多态等 OO 思想的特性等。这种题目需要较多的编程实践和对细节的把握。
- 第二种就像故事中的那道编程题一样，考查的是应聘者的逻辑思维能力和对算法的理解，希望各位同学在下工夫啃 Java 的时候，能分出精力来研究数据结构和算法方面的知识，那是不以任何语言和平台为转移的。

牛开复把卷子翻到最后，说道：“编程题我们就谈到这，最后我们来谈谈有可能出现的智力题，这种题目还是很锻炼思维的。”

题目：

- 有一辆火车以每小时 15 公里的速度离开洛杉矶直奔纽约，另一辆火车以每小时 20 公里的速度从纽约开往洛杉矶。如果有一只鸟以每小时 30 公里的速度和两辆火车同时启动。从洛杉矶出发碰到另外一辆车后返回，依次在两辆火车来回飞行直到两辆火车相遇。请问这只小鸟飞行了多长距离？
- 想象你在镜子前，请问为什么镜子中的影像可以颠倒左右却不能颠倒上下？
- 如果你有无穷多的水、一个 3 升的和 5 升的桶，你如何准确称出 4 升的水？

“师兄啊，我看到第一题就有些茫然了，这要算很久的吧？剩下两个我也没太想明白。”

“你是研究书本多了，思维变得有些僵化了，先看第一道题，如果换个角度考虑，小鸟飞行距离的另一种算法是用速度乘以时间，速度是已知的，而时间就是两辆车相遇所用的时间。于是这道题就变得非常简单了吧？”

“哦。那第二道题也应该是换角度考虑的吧？”

“没错，第二道题你可能会想到物理上的知识、平面镜成像啊等，不过这道题的正确答案是：如果把镜子放在天花板或地板上，镜子中的影像就会是上下颠倒的了。”

“第三道题呢？”

“第三道题很经典。答案是这样的：把 5 升的桶装满水，再倒入 3 升的桶直至其满溢，然后清空 3 升的桶，再把 5 升桶中剩下的水（2 升）倒入 3 升的桶中，最后往 5 升的桶中装满水，倒入 3 升桶中直至其满溢，5 升桶中剩下的水就是 4 升了。”

“哇，看来身为开发人员思维要活跃啊。”

“那是必需的啊，呵呵。蔡佳娃，经过今天的小试牛刀，感觉怎么样啊？”

“笔试还真不是个可以偷懒的活，不光要看书，更重要的是培养思维啊。”

“呵呵，你脑子还可以，肯定没问题的。我等你的好消息，好好考呀。”

“恩，师兄，我一定全力以赴！”



提示 作为一个开发人员，永远做的是思维的运动，在此建议各位读者把工夫下在平时，临时抱佛脚或许会有用，但厚积薄发来得更为爽快。

4.2.3 笔试小结

虽然历来考试高一尺，作弊高一丈，但是笔试这一关的蒙混几率却不是很高，反而会有一些“高手”因为平时对基础知识和一些小问题不太留意而 fail 掉笔试，与心目中的公司失之交臂。因此，作为初学者不可毛躁，应该在每次实践中注意细节，勤于积累。

4.3 面试——最难的 BOSS

有句俗语说得好：“是骡子是马拉出来溜溜”，闯过了简历关和笔试关，面试这一关就是拉出来溜溜的时候了。前面两关或许还可以躲躲闪闪、坑蒙拐骗，到这里可就是丑媳妇见公婆了。面对这个游戏中最难对付的 BOSS，该如何将其击败呢？

4.3.1 面试面什么

话说上次蔡佳娃在潜心向牛开复师兄请教笔试技巧之后，自己又很不放心地狂看了两天书，然后就杀气腾腾地去笔试了。后来他在跟师兄的电话里用“虽艰难，终涉险”来形容这次面试。现在，摆在蔡佳娃面前的就是这道面试关了。

“师兄啊，这个简历和笔试，或多或少我们在学校里也大概接触过，比如写报告和考试，可是这个面试我们可真是少有经历啊。后天我就得去面试了，师兄你给我支支招吧。”

“找工作这回事，越往后考查得越全面，越不大可能靠临时准备应付过关。面试主要看你有没有技术和非技术上的问题，是不是适合目标公司。”

“我听说面试也要看考官是谁，HR 和技术不一样吧？”

“恩，HR 考官主要看你这个人怎么样，比如你的谈吐举止、反应能力和思维能力、有没有进取心、有没有工作热情等。而技术考官则要看你的专业素质，比如会问一些技术上的问题、会针对你的简历提出相关的问题以检验简历的真实度等。”

不同公司 HR（人力资源）和技术考官的面试顺序一般有很大的不同。小一些的公司人力资源方面的因素在最终决定时占的比例相对较少，而大公司还是很重视人力资源方面因素的。

不过一般情况下还是技术考官，即应聘者将来的上级发表主要意见，如果技术面试通过，HR 的面试就是大致看一下应聘者有没有大的毛病。不过有时 HR 考官同时也是技术考官，也有些时候会只有技术考官的面试。

“师兄，那 HR 面试的时候会问些什么呢？”

“一般的 HR 都不是搞技术的，所以问的问题不会很专业，但是可不要小觑 HR 的本事，往往专业技能很全面的应聘者却会被 HR 看似犀利的问题逼得结结巴巴。”

“啊，我可不想被 HR 给整崩溃，师兄你说说他们会怎么提问我们啊？”

“最一般的情况他/她会让你做个自我介绍，或者讲讲自己的特长和不足，或者谈谈自己的职业规划，谈谈如果顺利进入他们公司你有什么打算，如何开展自己的工作，自己对薪水的问题是什么样的看法等。”

“天哪，这些问题还真的从来都没有想过呢。要不是听师兄你这么一说，只怕到时候被问到，也整个大红脸啥也说不出来，那样就惨了。”

“呵呵，别这么不自信。其实 HR 考官在问你这些问题的时候，一般情况下并不是真正想要听你的答案是否正确，而更多的是分析你的回答思路是不是清晰，口齿是不是伶俐，有没有条理，有没有过分的紧张和不安。有时候还会问一些很突兀的问题呢。”

“啊，这么夸张啊！”

“是啊，你要想想，身为 HR 面试考官，他们也需要创新和进步啊，所以不能总停留在那些中规中矩的问题上吧。比如他们会问你我们凭什么录用你？怎样对你你才会辞职？上级不喜欢你怎么办？你的家人不支持你的工作怎么办？想在公司混到什么职位？”

专家分析：尽管 HR 考官大都并不擅长技术，但是就算你技术再过硬，往往面对 HR 考官丢过来的问题也会有些接招不暇。不过，醉翁之意不在酒，在貌似与技术能力无关的问题背后，聪明的考官往往看中的是以下几点：

- 举手投足：从求职者进门开始，考官就在观察，每一个小动作都会反映求职者的性格特征，都会成为考官为其打分的依据。
- 交际口才：考官通过求职者回答问题的方式和内容，会对其表达能力和交际能力有一定的了解。这些能力虽然在技术上帮不上忙，但是须知 IT 行业可不是扛着技术大旗就可以打天下的。
- 工作态度：考官会从求职者的言语和举止中看出其工作态度是否认真、热不热情、积不积极、稳不稳重。
- 应变能力：考官会提问一些比较突兀的问题，通过观察求职者的回答延迟、回答流利与否、答案是否条理分明，来判断求职者的应变能力。

“牛师兄，我觉得 HR 考官有时候挺像记者的，提的问题很不好回答。”

“呵呵，HR 考官就是靠这些貌似很不好回答的提问来为自己的公司挑选合格人才的。真正的人才，不应该只是技术能力强，为人处事、交际、应变能力都是同等重要的。所以啊，不要小瞧

HR 考官，他们往往会从你手忙脚乱的表现中看出你的破绽。”

“让你这么一说，HR 面试这关也是很险咯？”

“HR 面试还不算最困难，因为其主要是为技术考官的面试提供一些参考意见，往往还是技术考官说了算呢。”

“那技术考官面试的时候会怎样出招呢？”

“技术考官一般没太多心思研究你这个人的性格特征，他们直接上来就问技术问题。所以到了那时候你也就别谦让了，有多大能耐是多大能耐，尽最大努力促使考官点头吧！”

在这里需要提醒一下，笔试中曾考过的题目，在技术考官面试这一关，是完全有可能会再次问到的。但并不是说，技术考官的面试就会稍微简单一些。如果笔试中某道题答得不错的话，在这一关同一道题的回答却可能让技术考官眉头紧皱。

即使是相同的题目，却是一说一写不同的答法，技术考官可比 HR 考官敏锐得多，你回答问题的思路清不清晰、对这部分知识掌握得透不透彻，他们马上就可以看出来。因为写在纸上的知识和嘴里说出来的知识绝对是不同的。

“恩，看来还是不能掉以轻心啊。”

“是啊，技术考官除了会问你这些问题，还会根据你的简历让你介绍一下项目经验的细节呢。比如技术考官会拿着你的简历问这些问题：

××环节具体是怎么实现的？具体采用了什么技术？

××项目过程中出过什么问题？怎么解决的？

在××环节有没有试着用这种方法来解决？为什么不用？

……”

“哇，师兄，我有点害怕了耶。”

“不用害怕，能够挺到面试，如果没有弄虚作假，都不会是等闲之辈，拿出自己的本事，只要相信自己能行，就不怕他们问。”

专家分析：技术面试是比较有决定性的环节，主要考查的是求职者的以下两个方面。

- 技术能力

项目是不是真做过，是不是真明白，是不是真掌握了，特长是不是真精通，是不是蒙混过关的，一问便知。所以这里也是对简历的一个检验。

- 思维能力

求职者对于技术考官提出问题的回答，便是其脑中思维的外在体现。任何一家用人单位都会想录用那些逻辑思考能力强、思维活跃的人才。

4.3.2 支招面试

前面一小节本书介绍了面试这一关中 HR 考官和技术考官所考查的人才的不同方面，本小节将继续跟随蔡佳娃的脚步备战面试。

“师兄，听你这么一说，面试，尤其是技术考官的面试，的确是来选拔人才的。看来这一关的确不是能混过去的啊！”

“那是当然，这算是求职过程中至关重要的一关，一面定江山呢，所以千万不可马虎。当然也不用过分要求自己。毕竟学问有高低，专业的思维能力也不是一天两天能够练成的。过分地准备或许会弄巧成拙，不过也不可打无准备之仗。”

“那师兄你给我指点一下迷津吧，不管怎么说我也得试试看。”

“恩，很好。首先看 HR 面试。其实与 HR 面试也可以算得上是聊天。HR 考官只是想了解你，看看你是不是适合人家的公司。所以你首先就是要做到不紧张，轻松一些。”

“哎，这一点我到了现场再慢慢安慰自己吧。”

面试的准备工作基本上该做的都应该在平时做完，比如技能水平提升、交际能力与口才的锻炼、个人性格态度的锤炼等。到面试那一天，只要把自己的形象好好整一整即可。

模样我们没办法优化，而着装和气质是可以后天打造出来的。不管结果怎样，都要面带微笑，以一种阳光、热情、诚实、有进取心的形象示人。拿出买卖不成仁义在的气势来，不要对自己定太高的目标，这样对 HR 的面试也是十分有利的。

“那具体一点说呢？比如 HR 考官来面试你了，该怎么应对啊？”

“呃，其实有些时候呢，HR 考官也像是在拷问囚犯，想方设法下圈套骗你说出自己的缺点，这时候就要注意，不可全部老实地作答提出的问题。很多问题都是有圈套的。比如考官问你有什么劣势，或者问你编程过程中经历过的最大的困难等。”

“我就撒谎，说没有？”

“不是，这个可以有。只是你不能太老实。应该巧妙地避开对你应聘工作不利的因素。让考官知道虽然你有一些缺点，也犯过一些错误，经历过失败，但还是不影响目标公司的岗位要求的。因此，适当地掩盖还是必要的。”



提示 考官在问问题时，都要注意到不仅自己的思维能力在被考验，自己的表达能力也在被考验，所以尽量不要让自己的嘴巴误了事。回答问题要干脆、斩钉截铁，不要吞吞吐吐，少说“呃”、“恩”、“这个”、“那个”等。

“恩，我明白了，还是不要过分牵强附会，有一说一是吧？”

“那是当然啊，人家 HR 哪个不是面过 N 张脸的人啊，你扯得太荒唐他们怎么会不晓得啊。尤其你再连说话都不连贯，就算你是真品也怕被人家鉴定成赝品呢。”

“哈哈，师兄说得很对，说话流利也是一种干练的表现。”

“是的，语言是思想的外在体现嘛！”

“那么，师兄啊，哪些地方我们还可以稍微做做假啊？”

“如果说还有的话，就是在 HR 考官研究你的时候，你也尽量分析分析考官的心理，尽量投其所好，在不恶性欺骗的基础上尽量让考官喜欢上你。比如考官是个女的，又问你关于家庭的问题，你就回答说会尽最大努力协调好工作与家庭的关系等。”

“如果考官是个明显的工作狂，我就说我一切为了工作，工作第一。家人一定会理解我、支持我的，对不对？”

“呵呵，学得蛮快的嘛，差不多就是这个样子。”

专家分析：面对 HR 考官的时候，本书给求职者的对策就是尽最大可能让考官喜欢上你，具

体需要注意的地方如下所示。

- 外在形象，海水是真的不可斗量，但人经常被貌相。所以注意自己的着装谈吐，让自己的合格形象打响面试求职的第一炮。
- 人格魅力，虽然谈魅力遥远了些，不过还是要尽可能让自己显得口才好、讲礼貌、有进取心、真诚、热情……当然如果这些特质读者并不具备，也不用过分伪装。
- 适当提升，毕竟 HR 考官不是技术考官，所以在技术层面的问题，还是可以夸张一些的。其他问题也要看情况而定，最好不要老实巴交地问什么答什么，说的话要为自己的目的服务。
- 做到无懈可击，面对 HR 考官丢过来的犀利问题，要学会避重就轻，避实就虚，尽量不让自己看出自己的破绽。
- 提前准备答案，对于一些提问几率很大的问题如自我介绍、优势劣势分析、如何开展自己的工作、为何辞去前一个工作等可以提前准备好。记住一定要尽量从用人单位的角度来考虑和准备问题答案，这样会使求职者在面对考官时更加成竹在胸。

“师兄你再说说遇到技术考官时应该怎么对付啊！”

“面对技术考官你就没必要装了，把自己所知道的、掌握的、精通的全都抖出来。还有啊，如果你的简历写得很高深以至飘渺，那就要小心了。”

“啊，难道考官还会拿着简历对口供吗？”

“是呀，要知道这可是一锤定音的环节，考官肯定要保证招聘进的人是真的可以做事的有用人才，养闲人可不是 IT 行业的习惯。”

“那我该怎么办啊？”

“这样吧，你回去准备一下，我们明天来一个模拟面试吧，只是技术这一块的，要不说了也是纸上谈兵。”

“恩，也行，像我这样的菜鸟最喜欢的就是举例子、打比方。呵呵。”

“你回去要做的就是：提前准备。自己做过的项目要提前做好技术讲解和项目架构图，到时候我问哪里你就答哪里。”

“项目架构图，那是什么啊？”

“就是项目功能实现的结构图示，你可以去网上搜一搜，试着画一下，明天我们模拟的时候我再帮你看看。”

“恩，明天见考官！”



提示 技术考官往往看人很准，尤其是看新人，所以在这一环节最好还是实事求是，好好准备一两个拿得出手的项目经验，这里毕竟不是学校。

4.3.3 面试演习

实践是检验真理的唯一标准，纸上谈了这么多兵法，也该拿到战场上运用一下了。提前感受一下兵书上的轻描淡写和战场上真实残酷的不同，或许对于一个新兵的成长是最有帮助的了。

“蔡娃娃，你准备好了吗？我们要开始技术考官的面试了。”

“师兄，不要手下留情啊，放马过来吧！”

演习开始！

“你好，请自我介绍一下。”

“啊，不是说是技术考官吗？怎么会问 HR 的问题啊，师兄？”

“你好，能自我介绍一下吗？”

“哦…呃…你好，我叫蔡佳娃，是××理工大学计算机专业大四的学生。我在大学期间……通过大学期间的学习和参加校外的培训，逐渐掌握了 Java Web 开发的一些知识……与技术，并且也参与了一些项目的工作，积累了一些经验。我对贵公司有一定的了解，结合自己的能力和经验，我觉得自己还是可以为贵公司尽一份力的。”

这是蔡佳娃这样的新人最容易暴露出的一个问题：想当然。虽然说了是技术面试，但是也不能想当然地将人力资源面试和技术面试完全区分开，制定本来就不存在的虚拟规则。同时，对于突兀问题回答得仓促结巴，也反映出应变能力差的一面。

读者要注意，用人单位在招聘技术人员时是十分重视应变能力的。因为世界上没有任何两个项目是完全相同的，作为一个合格的开发人员应变能力十分重要，读者朋友们平时要注意尽可能多地培养自己这方面的能力。

“恩，现在 IT 公司这么多，你为什么选择我们公司？”

“那个……贵公司对行业形势判断很好，而且我认为以自己的能力，是可以胜任这项工作的。另外，我选择贵公司也是因为这样离我的家近一些。”

“那么，你觉得如果你被录用，摆在你面前的最大困难将会是什么？”

“我觉得可能是工作环境的问题吧，毕竟在学校和公司的工作环境都不一样，可能要花些时间来适应吧。”

“你觉得你哪方面最不行？”

“我觉得我有时候太钻牛角尖，不达目的不罢休，可能会耽误一些正事的进度。”

这一部分蔡佳娃表现得很不错，除了那句“离我的家近一些”需要被列为毫无意义的废话，因为这句话跟面试无关而且也会让面试官觉得此人不够稳重和严谨。而关于自己不足的回答就很精彩了，既没有正面否定不谈，又讲了一些既像优点又像缺点的不足之处。

“呃，你的简历上面说你对 Java Web 开发掌握得不错，讲一下 Servlet 的生命周期吧。”

“恩，Servlet 是在 Web 服务器启动的时候 new 出来的，然后如果有 http 请求到来时，就会调用自己的 service()方法，根据请求的不同执行 doGet()或 doPost()方法。Servlet 是可以同时处理多个请求的，当服务器停止的时候，Servlet 也会被销毁。”



提示 关于 Servlet 生命周期这个问题，蔡佳娃回答得也不错，在谈到技术的时候，最好少用口头语，尽量用技术用语，而且语言越精练越好。

“好的，说说你做过的项目吧。”

“我在大三的时候参与过一个移动环境监测系统项目的开发。”

“是吗？请讲一讲项目的总体功能吧。”

“好，这个项目主要是实现对环境监测点的环境质量进行上报和查询，监测的内容包括水质、大气和违法排污现象。分为手机端和 Web 服务端。”

“具体手机端和 Web 服务端的功能是怎样的？”

“手机端主要负责大气和水质情况的上报和查询，Web 服务端具有手机端功能的同时，还作为服务器具有接受上报数据、发回查询数据、管理数据库等功能。”

(递过去一张纸)“请你画一下项目架构图吧。”

“啊，不用，我带着呢。(见图 4-1)”

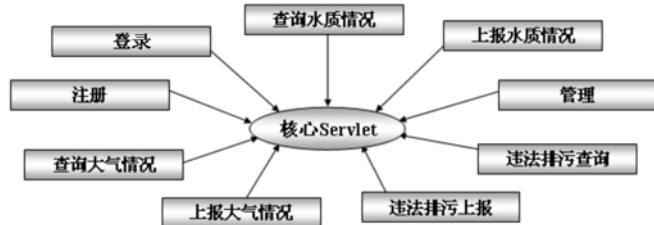


图 4-1 蔡佳娃的项目架构图

这里又暴露出一个问题，首先不说蔡佳娃画的这个项目架构图是好是坏，他不应该将一张已经画好的项目架构图递过去，因为在这种情况下面试考官是希望蔡佳娃自己现场画的，这样还能考查一下这个人的知识再现能力。

“啊？你这个图……，好，你详细说一下你在这个项目中都做了什么吧。”

“我主要负责 Web 服务端的设计和开发，这个项目中要有两个 Servlet，一个负责处理手机端的请求，一个负责处理 PC 浏览器端的请求。”

“那服务端是如何接收手机端请求的呢？”

“首先在 Servlet 的 doPost()函数中调用 request.getInputStream()函数得到请求的输入流，并将其封装到 DataInputStream 对象中，再调用 DataInputStream 类的 readUTF()函数获得字符串格式的请求数据。”

“服务端是如何响应手机端请求的呢？”

“Servlet 通过调用 response.getOutputStream()函数得到输出流，并将其封装到 DataOutputStream 对象中，再调用 DataOutputStream 的 writeUTF()函数发送响应数据。”

“开发的过程中有没有出现过什么值得留意的地方呢？”

“有，因为要面向手机端，而不同的手机平台对 Java ME 的支持有细微的差别，比如在面向的手机终端为诺基亚时，要在每次向手机端发送数据时在 writeUTF()函数后面调用 flush()函数，否则手机端就不会收到响应数据的。”



提示 对于项目中值得留意或收获最大的地方这种问题，如果实在没有就不要乱说，因为技术考官都是上过战场的老兵，不恰当地班门弄斧，最有可能给自己画蛇添足。

“你觉得这个项目的缺陷在哪里？”

“我觉得界面还是不够友好，全是数据啊，表格啊，客户可能看了不会太喜欢。”

“如果要你再优化一下这个项目，你会怎么做？”

“我想在 Web 端用 AJAX 技术做一个环境质量监测图，这样视觉效果会好很多，同时手机端也可以用 GameCanvas 实现类似的视觉效果。”

“那么，谢谢你的作答，我们会给你电话的。”

“好的，谢谢你，再见！”

演习结束！

总体来讲，蔡佳娃的这次面试初体验还是比较成功的。除了开始的几个貌似应该归到 HR 面试题中的问题，蔡佳娃在技术层面上的回答很简练准确，这种比较干脆的表达方式是比较受技术考官欢迎的，看来准备充分些的确是会有些优势的。

“师兄，怎么样？我表现得还行吗？”

“我先不说，你觉得自己表现怎么样？”

“我觉得前面那几个问题回答得不是很好啊。”

“那不是一般地不好啊，我昨天跟你说的避免犯的错你基本上都犯了呢。”

“啊，不会这么夸张吧。不过，师兄你说话不算话嘛，明明说了是技术面试，怎么还会有自我介绍之类的问题啊。”

“你们就是在学校呆得太久，总想把所有事都摸索出一个固定的规律来，然后再按照规律制定一个固定的策略。求职游戏可不像八股文那样有这么多条条框框，它是很自由的。所以如果你只是按照固定的模式去准备，而不想着如何应对突发情况，那就有风险了。”

“对，现在已经不能像在学校考试那样押题或是猜题了。”

“就是嘛，你看你的回答，又是口齿不清，又是跑题，问你为什么选择人家公司你居然还会说因为离家近。”

“哦……我那时有点懵。”

“还有，问你可能最大的困难是什么，虽然那个困难不算什么，但是你说话的语气却让考官我很担忧啊，人家还以为你在担心自己没办法适应新环境呢。”

“哎，看来要注意的问题还真不少啊。那其他的不足呢？”

“其他问题回答得都还可以。下面我们看看你的项目问题。你这个项目还算不错，不过你看看你的项目架构图，画的叫什么啊。要是没有 Servlet 这个词，人家都不一定会认为是个软件项目的架构图呢。我把我当年找工作时用到的项目的架构图给你看看。（见图 4-2）”

“这是一个学生信息管理系统的项目架构图，写得比你的要详细多了吧。”

“不过，除了比我画得多以外，其他也没有什么区别吧？”

“区别可大了！首先你的图画得太不具体，只是把功能模块列出来了，那根本不是架构图，只是功能罗列图。架构首先是你的项目的详细物理结构，所以应该是由一个个项目文件组成，每个项目文件会执行一个或多个功能，数据流向就说明了这些功能的执行结果和去向，而且也没有数据库模型。”

“真的是好复杂啊，当时师兄你找工作的时候就是带着这个图去面试的吗？”

“怎么会带着呢。你想得太好了。都是给一张纸，现场画的，所以我刚才看到你居然拿出一张图来就很诧异了。”

“啊，不是吧，这么复杂怎么能画得全呢？”

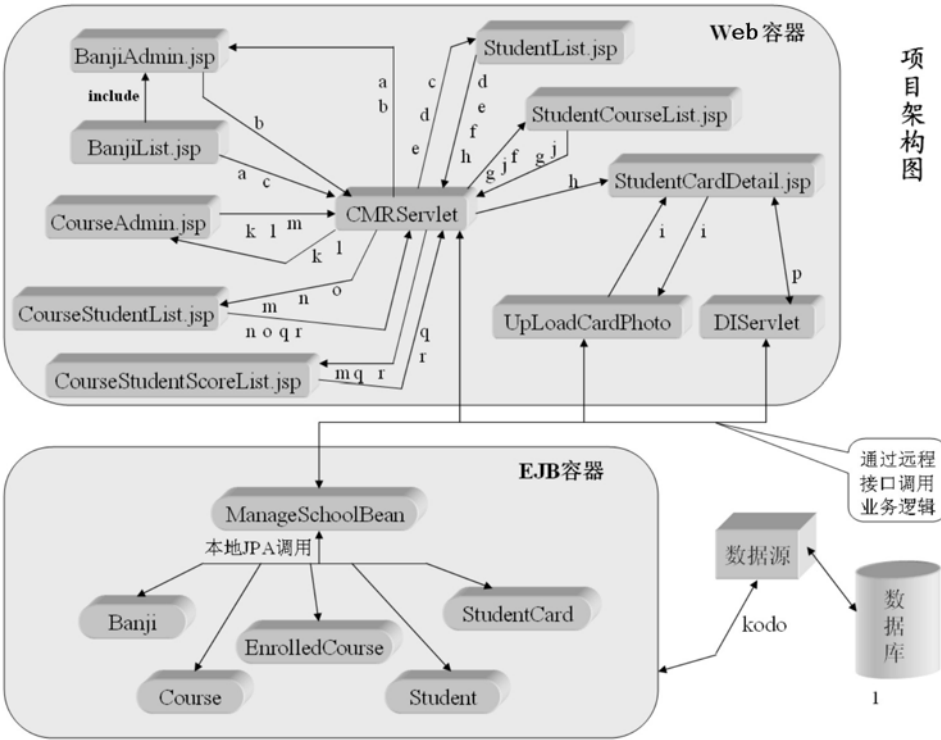


图 4-2 学生信息管理系统架构图一

“这个图给你看是仅供参考的，真实情况下并不需要全部画出来的。你不要把顺序搞错了，项目架构图是项目开始之前画出来的，之后的项目开发都以此为据。而不是把项目写完了之后总结出来的。”

“啊，幸亏有师兄你提醒啊，我差点就搞错了呢。还有啊，师兄你这个项目架构图上的 abcd 等字母是干什么用的啊？”

“这个是动作编号，是执行的具体功能。这是对照图。（见图 4-3）”

动作编号对照表

动作编号	动作内容	动作编号	动作内容
a	删除班级	i	替换照片
b	添加班级	j	删除学生选课
c	查看学生明细	k	添加课程
d	添加学生	l	删除课程
e	删除学生	m	查看选课学生
f	查看所选课程	n	添加选课学生
g	从学生删除选课	o	删除选课学生
h	查看学生证	p	获取照片
q	提交成绩	r	暂存成绩

图 4-3 学生信息管理系统架构图二

“哦，那当时师兄简要地把这个项目的架构图画出来后，考官是怎么问你的啊？”

“当时考官问我这个项目我负责的部分，我说我负责的地方就行了。”

“哦，我记住了。师兄，有了这次演习，我就有一些底了，就等明天一面定江山了。呵呵。”

“OK，师兄等你好消息！”



提示 在技术面试之前把简历中提到的项目准备一下项目架构图，对自己负责模块的技术细节问题进行一些回顾是很有必要的，免得在实际面试时打无准备之仗。

4.3.4 面试小结

本书虽然是先谈笔试后谈面试，但是在真实情况下，笔试面试的顺序可能有所不同。而且笔试和面试没有一个统考大纲，各个公司也不会抽时间聚到一起来研究如何招聘人才。因此，本书介绍的是大众情况下的求职之路。



提示 有些情况下不仅笔试面试的顺序不同，笔试和面试的内容也会与本书所介绍的有所不同。而且，说到底，面试是个纯主观的项目，所有的题目（技术题目除外）都不会有唯一标准的答案。所以希望广大读者还是以充实自己第一，谋攻策略第二。

4.4 试用期——这才是最后一关

如果各位读者认为自己已经被“拉出去溜了溜”，而且顺利打败了游戏中最难的 BOSS，以为高枕无忧那就错了，试用期才是这个游戏中的最后一关。如果前面的几个关卡还主要注重的是竞争和考试，那这一关就是主要看实践了，实践是检验真理的唯一标准。



提示 试用期这一关并不是最难的，但却是最重要的，每一位求职者都不想功败垂成，所以，平安度过或者靠自己能力缩短试用期，就是步入 IT 这个江湖的最后一跃。

4.4.1 试用期考查什么

每一位进入试用期的求职者都是踌躇满志、血气方刚的。但是，经过了前面几关的洗礼，到了这里仍然不可以放松心态。试用期时一只脚已经踏入了 IT 这个行业，所以试用期所做的一切，全都是为了外面的那只脚。

蔡佳娃到了面试那天居然人品爆发，面试结束后立刻就接到电话宣布进入试用期。蔡佳娃当然不忘感谢一直帮助他的牛开复师兄，马上打电话给师兄要请他吃饭。当然，在饭桌上，蔡佳娃也不免要向师兄咨询一下传说中的试用期。

“师兄，你说试用期是个啥玩意啊？”

“首先呢，试用期可长可短，完全取决于你在公司的表现。其次，试用期可以升级为就业合同，也可以走向下次求职。”

“那师兄照你这么说，试用期也不是那么保险的呗。”

“看看，你又想放松了是吧。没有什么事情是万无一失的。况且 IT 这个行业又是个技术更新快、瞬息万变的行业。”

“是是，师兄说得对。那试用期公司考查我什么呀？”

用人单位判断是否让一个试用期的人转正的条件有如下几个，这也是用人单位在试用期重点考查的方面：

- 第一个依据是此人是否名副其实，像他简历所说的那样，该精通的知识都精通，是否有弄虚作假的嫌疑。所以如果前面几关水分大，到了这里就有可能露出狐狸尾巴。
- 第二个依据是此人是否为公司创造了价值。公司存在的唯一目的是盈利，说别的都是套话。所以如果一个人没有为公司创造价值，那这个人就可能要为公司节约一些财富，即被辞退。现实很残酷，所以在试用期要拿出百分之二百的努力来拼。
- 第三个依据就是此人的人品和态度怎么样，对工作是不是负责，是不是有工作热情，是否诚实可靠，跟上级和同事关系怎么样等。
- 第四个依据就是此人是否具有一定的不可替代性，当然这里的不可替代性并不是指没了某个人公司就不能运转。而是根据你现在为公司创造的价值，公司得计算着辞退你再另聘他人和留着你继续培养哪个更合算。如果是后者，那么你就具有一定的不可替代性。

“师兄，这几个条件虽然听起来很残酷，但是道理却真是这样啊。”

“是啊，IT 行业可不是宫廷乐队，所以休想‘滥竽充数’。”

“师兄你说得没错，我在试用期内一定全力以赴，尽快转正成为一名真正的开发人员！”

“恩，全靠你自己了，放心大胆地使唤自己吧！”



提示 考验无处不在，没有永远安逸的地方，“生与忧患，死于安乐”。既然走到了这一关，就要继续坚持下去，功败垂成是最遗憾的。就像故事里说的那样，公司存在就是为了盈利，是否能为公司创造价值，就是衡量各位求职者能否转正的关键。

4.4.2 多做什么，少做什么

既然试用期也是如此“险恶”，蔡佳娃不禁心里发怵。他十分不想自己已经迈过去一只脚还要被踢出来，于是吃过饭后还是缠着牛开复师兄不放。

“师兄啊，明天就是我试用期的第一天上班了。你不给我点意见吗？”

“让自己安然度过试用期很简单，想想自己从前上小学是如何积极表现争取拿小红花的，原理大概差不多。”

“啊，师兄你太会开玩笑。”

“呵呵，在试用期内，你的底线是要保证积极按时完成上级交给的任务，如果这个都做不到，那真的与小红花无缘了。”

“恩，那个的确是根本。其他方面呢？”

“试用期内你应该注意多做些什么，少做些什么，让自己尽快成长为一个开发人员。”

在试用期内，要多看、多听、多做。从学生变成开发人员，要学的东西很多，而一般也不会有人专门为你搞个培训，所以只好自己偷着学，通过观察其他人，慢慢找到开展工作的门路。剩下的就是靠自己的专业技能为公司出力了。

同时，要少说空话、废话、无用的话，少唱高调，要明白自己已经不是一名大学生了，少把自己大学时逃课等坏习惯带到公司来。一个新人，入行的形象最好是低调、踏实、肯干，所以应

该尽量避免让自己哗众取宠。

“对对对，是该低调一些，何况我还是个菜鸟。”

“其他方面，你在公司应该不怕脏、苦、累。多干活，少埋怨。甭管分配的工作喜不喜欢干，先拿下再说。展现你的工作热情和旺盛的战斗力和战斗力，这可是一些老员工所没有的。而且我刚才说的低调可不是有新创意憋着不说，虽然我们是新手，可是该出手时就出手，风风火火闯九州嘛。”

“呵呵，师兄，你说得很正确嘛。我会好好做个菜鸟的，当然不会做太久，我要立志做个像师兄这样的高手！”

“哈哈，努力就一定可以做到。师兄我也没有那么厉害啦。我们共同进步！”

4.4.3 试用期小结

试用期是求职的最后一步，也可以说是真正上班的预备期。虽然还不是正式员工，但是也要拿出比正式员工还高的热情来，主动出击而不是坐以待毙。既要初生牛犊不怕虎，也要小心驶得万年船。最重要的还是用自己的专业技能为自己赢下一份满意的正式用工合同。

4.5 本章小结

本章向读者介绍了求职过程中所经历的各种关卡和具体攻略。希望能对广大读者的求职之旅有所帮助，求职是每个人步入职场的第一步，摆正心态，认清自我，确定目标，不骄不馁，昂首前进是本书对广大读者的最好建议。



提示

尽管本书提出了种种应对策略，但是各位读者要明白，蒙混过关不可取，不然就会让自己总是徘徊在躲初一和躲十五之间。真才实学才是硬道理，做最好的自己就是成功。



Java程序员 职场全攻略

——从小工到专家

吴亚峰 著

懵懂菜鸟如何才能成长为一名成功的开发人员？

如何在“血雨腥风”的求职大战中脱颖而出？

职场中只要技术过硬就可以吃遍天下吗？

是学招式重要，还是练内功重要？

掌握了哪些技能，才算是真正的高手？

.....

资深技术专家，娓娓道来。



 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>