TURING 图灵程序设计丛书

[美] John David Dionisio Ray Toal 著
贾洪峰 李松峰 译

JavaScript 程序设计

Programming with JavaScript
Algorithms and Applications for Desktop and Mobile Browsers

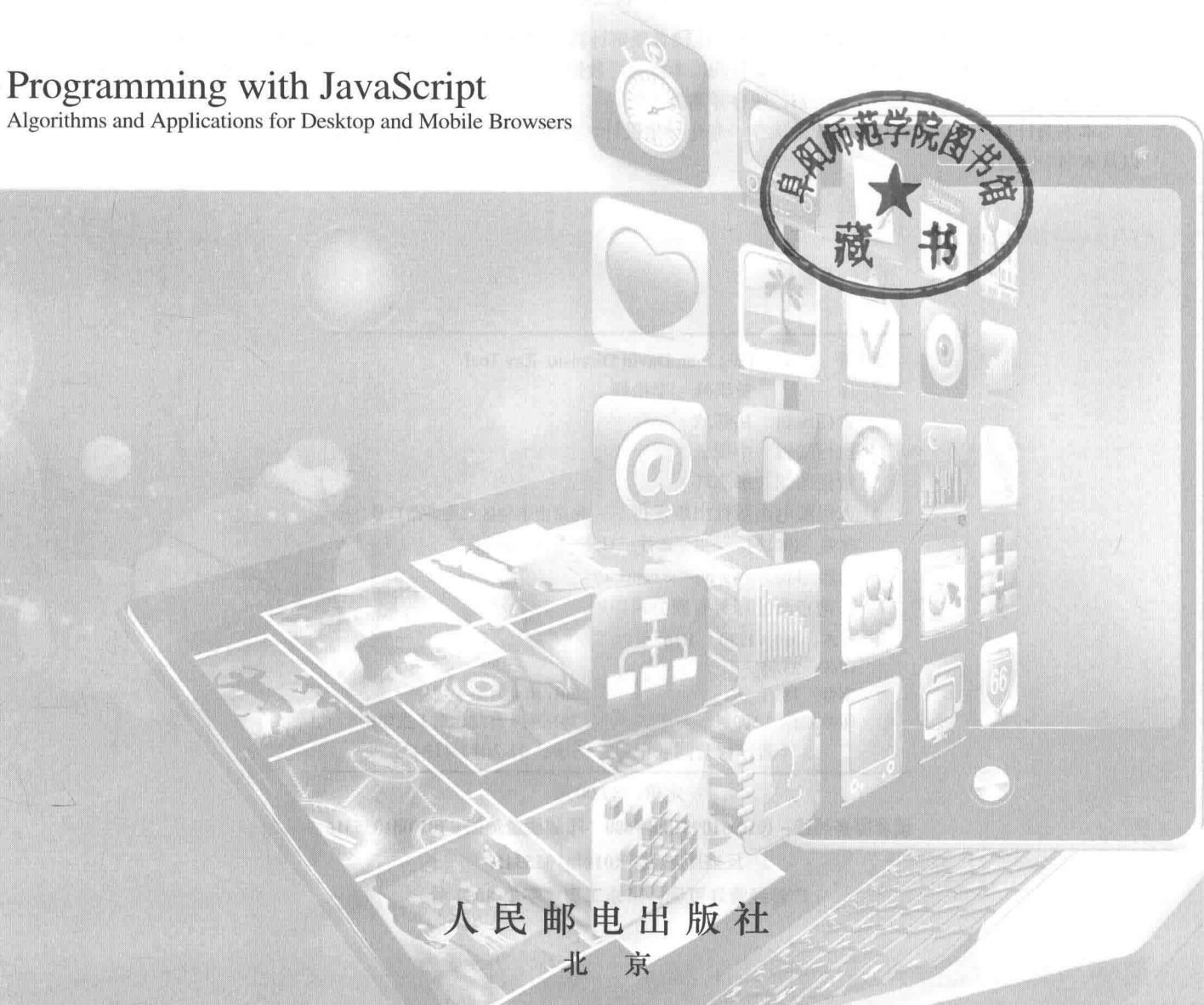
中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

[美] John David Dionisio Ray Toal 著
贾洪峰 李松峰 译

JavaScript 程序设计

Programming with JavaScript
Algorithms and Applications for Desktop and Mobile Browsers



人民邮电出版社
北 京

图书在版编目（CIP）数据

JavaScript程序设计 /（美）迪奥尼西奥(Dionisio,J.D.)，（美）托尔（Toal,R.）著；贾洪峰，李松峰译. -- 北京：人民邮电出版社，2016.4
（图灵程序设计丛书）
ISBN 978-7-115-41816-6

Ⅰ. ①J… Ⅱ. ①迪… ②托… ③贾… ④李… Ⅲ. ①JAVA语言－程序设计 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2016)第038958号

内 容 提 要

本书旨在通过从零开始介绍 JavaScript 编程让读者理解计算机科学的基本思想和原理。书中内容丰富全面，阐述由浅入深。主要内容有：计算的相关知识、编程的基本概念、数据、语句、函数、事件、软件架构、分布式计算、图形与动画，此外还探讨了正则表达式、递归、缓存等高级主题。

本书为计算机科学或软件工程专业大学一年级学生设计，高年级学生或新接触 JavaScript 的专业程序设计人员也可以从本书中获益。

◆ 著　[美] John David Dionisio　Ray Toal
译　贾洪峰　李松峰
责任编辑　岳新欣
执行编辑　牛现云
责任印制　彭志环

◆ 人民邮电出版社出版发行　北京市丰台区成寿寺路11号
邮编　100164　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
三河市海波印务有限公司印刷

◆ 开本：880×1230　1/16
印张：23.75
字数：785千字　2016年4月第1版
印数：1－2 500册　2016年4月河北第1次印刷
著作权合同登记号　图字：01-2014-4184号

定价：89.00元

读者服务热线：(010)51095186转600　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广字第8052号

版 权 声 明

感谢 Mei Lyn、Aidan、 Anton 和 Aila 一直以来对我的支持。我永远爱你们。

——JDND

前　言

听到编程和计算机科学这两个词，你脑子里会想到什么？爱玩游戏、不善交际的极客？或者计算机？这些当然都是自然而然会产生的意象。但实际上，任何人都可以编程[①]，而计算机科学涉及的内容也远不止计算机。除了给计算机编程序之外，你很可能还见过人们给手机、机器人、导航系统和工厂的机器编写程序。

本书主要介绍计算机科学的基本思想和原理，会涉及软件工程和信息技术的相关学科。我们希望你在掌握基本编程技能的基础上，来理解这些原理。计算机科学涉及面很广，包括计算、算法、软件系统、数据组织、知识表示、语言、智能和学习等。然而，只有通过实际编程，才能更好地理解这些概念和研究它们的工具。

本书目标

本书的目标如下。

- 介绍计算这门自然科学，包括计算机科学、软件工程、计算机工程、信息系统以及信息技术。
- 澄清关于计算的诸多误解，告诉大家计算是诸多领域职业发展的基础，包括医药、法律、商业、金融、娱乐、艺术、教育、经济、生物、纳米技术和游戏。
- 尽早培养大家对编程审美、标准、风格约定和审慎注释的意识，旨在把坏习惯扼杀在萌芽中。
- 通过讲解过去那些不会让初学者接触的难点，告诉大家 JavaScript（相对其他语言）的巨大威力。本书中某些这样的高级内容都加了星号（*），有的包含在附录里。
- 澄清一个问题，即编程并不是简单地把程序写得能运行、不出错，还要考虑如何编排更容易让人看懂，更容易修改，并且运行效率更高。
- 给出几个研究案例，包括分布式计算、手机或平板电脑等触摸界面，以及图形方面。让学生可以找到工作，让专业人士与时俱进。

本书内容

本书内容由浅入深，可以从头到尾依次阅读。我们其实是在讲一个关于计算和编程的故事，特别是 JavaScript 编程，主要内容简介如下。

- 第 1 章介绍计算这个领域的知识。
- 第 2 章到第 8 章介绍（JavaScript）编程的理论和实践。

  第 2 章讲解编程的基本概念。

  第 3 章讲数据。

  第 4 章是微观编程之一：语句。

  第 5 章是微观编程之二：函数。

  第 6 章是微观编程之三：事件。

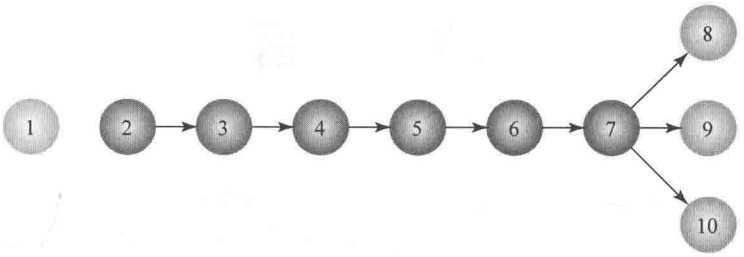
  第 7 章讲宏观编程之一：构建软件系统。

① “任何人都可以”的意思是优秀的程序员不受背景和经历限制，而不是说不用努力就可以[Bra07]。

第 8 章讲宏观编程之二：分布式计算。

❑ 第 9 章和第 10 章探讨高级主题。

虽然读者并不需要从头到尾依次阅读各章，但还是有必要告诉大家章与章之间的依赖关系，如下图所示。



注意，第 1 章是独立的。如果读者想直接看关于编程的内容，可以跳过第 1 章。

读者对象

本书是以计算机科学或软件工程专业大学一年级教材或主要参考书为目标设计的。要学习本书内容，不需要有编程经验。不过，它对初次接触 JavaScript 的高年级学生和专业程序员也是有价值的，因为本书并没有回避这门语言的难点或“高级”内容。事实上，我们相信本书中大量的思考题和随处可见的练习，以及关于 ECMAScript、HTML、Ajax、jQuery、图形与动画等现代 JavaScript 主题的讲解，即使对专业程序员也是大有裨益的。

JavaScript

这里要跟老师说两句：我们满怀信心地选择 JavaScript 作为培养新一代计算机科学家的语言。以前，JavaScript 很少被选为大学计算机科学课程的入门语言。这或许是由于各方面的误解吧[Cro01]。但我们认为，JavaScript 是这个层次课程的理想之选。

首先，由于 Web 浏览器无处不在，几乎每个学生都可以轻而易举地找到 JavaScript 解释器，无须下载和安装。其次，有些教授认为学生不应该上来就动手写代码，而是应该先通过伪代码来学习抽象的算法，但另一些教授又认为只有通过真正手写代码，才能让学生牢固地掌握这些概念，那么 JavaScript 恰好是这两种意见的一个折中。JavaScript 的语法清晰、简单，学生几乎可以立即上手，而不必先搞清楚类、public static 方法、神秘的 `void`、控制台、包等概念。我们发现很多学校为计算机科学大一学生选择的“简单语言”是 ML、Scheme、Ruby 或 Python。但随着 Web 作为应用运行平台的兴起（包括桌面和移动端），这些语言的流行程度没有任何一个可以与 JavaScript 比肩。

最后，函数式编程在过去很长一段时间只有学院派计算机科学家才感兴趣，但到了如今的多核和大数据时代，这种编程方式正变得日益重要。此时，把 JavaScript 选为教学语言意义非凡。JavaScript 中的函数式编程对刚刚入门的学生而言相对容易理解，特别是与那些“括号套括号”或依靠块、连续体（continuation）、生成器等特殊构造的语言相比，简直容易理解太多了。

随书资源

访问 go.jblearning.com/Dionisio 可以找到本书每章末尾练习的答案、源代码、PPT 教学讲义、勘误，以及本书未收录在内的其他有用资料。

致谢

我们想感谢 Turn Media 公司的 Loren Abrams、克莱蒙研究大学的 B. J. Johnson、洛约拉马利蒙特大学的 Philip Dorin、罗彻斯特理工学院的 Daniel Bogaard、俄勒冈大学的 Michael Hennessy 和韦尔斯利学院的 Laurence Toal，感谢他们认真审读前期书稿并提出建设性意见。感谢 Kira Toal 和 Masao Kitamura 提供图片，感谢 Jasmine Dahilig、Tyler Nichols 和 Andrew Fornery 帮忙收集配套资料。当然，对于出版方 Jones & Bartlett Learning 员工给予的大力支持，我们同样心怀感激，包括高级策划编辑 Tim Anderson、项目编辑 Amy Bloom、产品总监 Amy Rose，没有他们专业又敬业的工作，本书不可能问世。此外，我们还想感谢 Caskey Dickson 和 Technocage 公司为我们的跨端脚本提供托管服务，没有他们则无站可跨。

目　录

第1章 计算的概念

计算是一门处理信息的学问。它解决的问题包括“什么是信息”以及“怎么编码、处理、存储和传输信息”等。计算科学家本质上研究的是信息处理，并通过实验来更好地理解这个处理过程。计算科学中的实验部分，包括人造计算设备的构建与程序设计。

本章概述计算概念，介绍这个领域中的各个学科，研究它的人可能会从事什么职业，以及关于这个领域的一些流传甚广（也让人很遗憾）的误解。学完本章，希望你能理解什么是计算，并产生继续学习下一章的的兴趣，我们将从第 2 章开始学习 JavaScript 编程。

1.1 计算是一门自然科学

近 20 年来，人们已经意识到计算并不仅仅局限于人造的计算机器，并且已经从自然和社会的角度，对计算过程，即系统运行时遵循的一系列规则进行了考察[Den07]。

- 生物的信息在其 DNA 中被编码，以及复制、转录、翻译这些信息的生物学过程，被称为分子生物学的中心法则（见图 1-1）。
- 化学反应遵循不同的规则进行。比如氢和氧的混合，就有 $2H_2+O_2 \rightarrow 2H_2O+E$。
- 社会结构和社区的演进会受到环境、政府及其他因素的共同作用。
- 可以把金融市场看成一个计算系统，这个系统会对买卖指令、恐惧惊慌和市场调控措施作出反应。
- 神经过程也是计算性的。在生物神经网络中，神经元（从其他神经元）接收输入信号，然后根据这些输入（向其他神经元）发送输出信号。

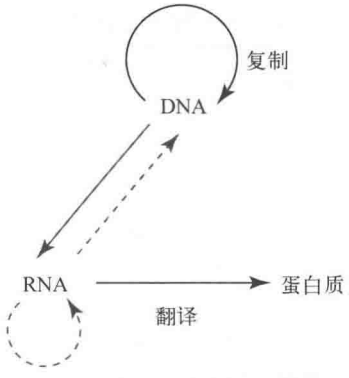


图 1-1 分子生物学的中心法则[Cri70]

研究自然界中的这种信息处理过程， 可以帮助我们更好地理解身边的这个世界。我们观察这些过程，记录下自己的发现。我们建造计算设备来辅助对这些发现建模。我们基于这些模型做实验，换句话说就是编写计算机程序，以期得到一系列“假如……”问题的答案。我们通过为企业、组织、政府和个人构建信息系统，来应用对计算的研究成果。

1.2　计算的五大学科

今天，很多人都认同计算研究领域有五大学科。

1.2.1　计算机科学

计算机科学涵盖一系列理论与实践，包括计算问题、算法、软件系统、数据组织、知识表示、语言、智能和学习。计算机科学家寻找如下问题的答案，并将其付诸实践。

- 什么可以计算，什么不可以计算？
- 某些计算可以用多快的速度完成？
- 要完成某些计算，需要保存多少信息？
- 如何高效、安全地编码、存储和检索信息？
- 如何设计信息处理过程（程序）？
- 我们怎么知道自己的程序没有问题？
- 计算理论怎么帮助解释智能和意识？

计算机科学关注计算本质及其在其他研究领域的应用。与哲学和数学类似，与教育某种程度上也类似，计算机科学把知识本身当作研究对象，而不仅仅把它当成研究手段。信息处理过程是通过编程语言表达的，而编程语言由语言学领域的概念来定义。计算方法可用于研究生物、社会和经济体系。人工智能作为计算机科学的一个分支甚至包含心理学的要素：解决问题的计算机给出的回答要想让人信服，必须展示其推理过程，而且这个过程人类必须认同。

1.2.2　软件工程

软件工程涉及设计、组织和构建软件系统（通常是大规模、至关重要的软件系统），重点关注生产效率、可靠性、健壮性、测试、维护和投入产出（见表 1-1）。软件工程分析业务需求，并通过设计软件来满足这些需求。设计软件既包括编写小段的代码，也包括把多个子系统集成起来形成一个大系统。软件工程师，也被称作*开发者*，在一个协作的团队环境下工作。

表 1-1　设计良好的软件应具备的特点

所谓软件……	意思是……
正确	能够完全按指令行事
可靠	不崩溃（就是不会意外停止运行）
高效	在合理的时间内完成任务，而且占用的空间也不多
可理解	用户和开发者能明白它为什么能自己干活
可重用	基于组件构建，这些组件又可以用在其他系统中
可扩展	可以扩展到处理更大规模的数据，而因此增加的成本有限
可维护	可以很快隔离和修复 bug（错误）
可用	能按照用户的想法和期望行事
经济	按时完工且在预算之内上线运行

1.2.3　计算机工程

计算机工程旨在设计数字系统，比如通信网络、计算机、智能手机、数字音视频播放器和录音机、显示器、汽车和飞机导航系统、警报系统、X 光机，以及激光外科手术工具等自动化器械。很多计算机工程师都是电子工程师出身，但专注于电子和数字系统，而且通常倾向于研究计算机硬件和软件。

1.2.4 信息技术

信息技术（Information Technology，IT）这个行当里的人会负责某个组织中的计算设施的建设、维护与故障排除，这些计算设施可能是计算机、网络、电子邮件系统、网站、数据库、电话及类似系统。除了编程，IT专业人员经常要完成复杂的配置、定制和升级任务。

1.2.5 信息系统

信息系统（Information System，IS）主要关注“计算方案”的设计，比如为公司、非营利组织、教育机构和政府部门更好地完成工作和提高效率设计信息化的解决方案。与其他四个学科不同，很多教育机构都会在商学院里开设信息系统这门课。

信息系统对数据库、通信工具等计算技术的使用研究较多。虽然编程也很重要，但与其他学科中的编程相比，重要性还是略逊一筹。信息系统专家更多的是使用电子表格，即通过嵌入在数据表中的公式来完成计算。图 1-2 中的单元格 E8 中保存的值就是一个公式：`=SUM(B8:D8)`，意思是要显示单元格 B8 到 D8 中各数值的和。

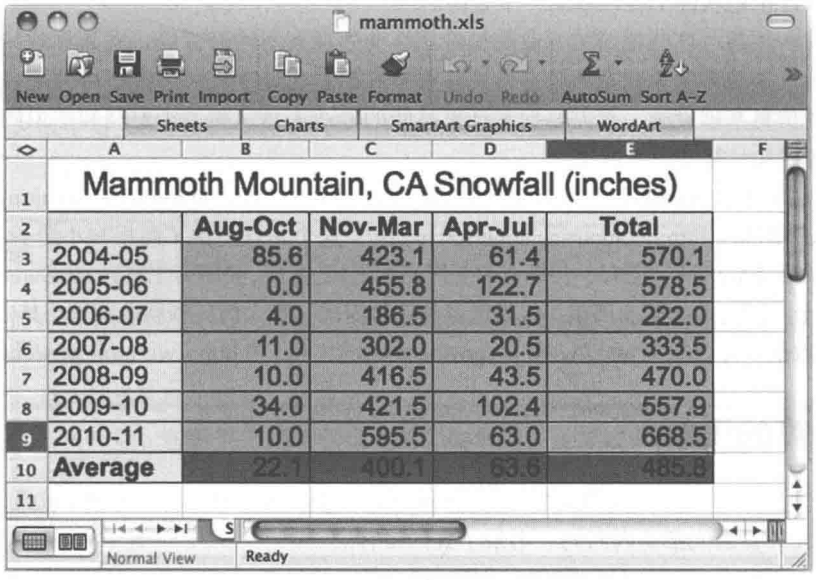


图 1-2 电子表格

1.3 与计算相关的职业

计算是一个宽泛的研究领域，因此掌握其中某个或某几个学科的知识，可以在以下行业找到工作。

- 生物。随着遗传学和计算生物学越来越重要，生物科学领域越来越有数字的味道了。很多生命过程都可以建模为信息处理。计算机科学家和生物学家为研究生物数据库和其他目标，经常需要合作。
- 搜索、数据挖掘和信息检索。很多人希望只通过模糊的搜索条件，就能很快从海量（且很大程度上并不相关的）数据中提取出自己想要的信息，而且愿意为此付钱。从大数据中快速提取信息一直都是计算机科学研究的重点。
- 娱乐。除了动画片，计算机在电影和电视领域也一直扮演着重要角色。大多数娱乐形式的制作过程，哪怕是现场演出，都要涉及很多硬件和软件。懂计算技术的人在这个领域非常有前途。
- 数字媒体。开车上图书馆或商店借阅或购买影片、音乐的人越来越少了。拥有计算背景的专业人士正在研究如何通过日益拥挤的互联网可靠、安全地交付流媒体数据，研究与版权和版税相关的问题。

- 游戏。游戏是产值数十亿美元的行业。很多玩家都喜欢追新游戏，因此也有很多程序员以开发和制作游戏为生。开发一款受欢迎的游戏并不容易，它是计算机图形、建模、算法，甚至数学、物理学、心理学和创意文案等多种技能结合的产物。
- 移动应用。很多过去安装在计算机硬盘上的应用，现在都可以从远程服务器随需下载。与此同时，移动应用的时代也已经到来，这些应用基本上都需要利用设备的位置信息（通过 GPS 系统获取）。而与这些移动设备和移动应用交互的新方式，需要计算专业人士来创造。
- 纳米技术。纳米技术就是设计和使用（数量巨大的）原子大小的设备，达成一种人类能够看得见的效果。这些原子大小的设备都需要复杂的编程。计算机科学在纳米技术领域应用的文章层出不穷[MS03，Mac09]。
- 网络安全与防护。机构（甚至国家）信息基础设施的巨大价值，意味着在可见的将来，对懂安全和加密技术的专业工作者的需求会一直存在。要具备这些专业知识，拥有计算背景非常重要。毕竟，加密和解密可都是计算过程啊。
- 航空航天。现代飞行器、卫星、空间站和太空探索机器人，每纳秒都要执行很多计算任务。这些任务非常复杂，可能需要传输数十亿字节的数据，而且对时间要求很高，甚至要求实时处理。这些大型复杂系统的设计正是软件工程领域的一个方向。
- 商业、法律和医药。这些领域的专业院校希望毕业生拥有不同的背景，尤其喜欢具有很强逻辑和分析能力的学生。医药影像学和商业信息系统从很早开始就与计算机科学密不可分了，保健服务提供商及保险公司的信息基础设施，都需要拥有计算技能的专业人员。而关于专利和知识产权的问题，则必须同时利用法律和计算知识来解决。

关于更多计算相关职业方向的信息，可以参考美国计算机学会（Association for Computing Machinery，ACM）的 Computing Careers 网站：http://computingcareers.acm.org/。这个站点提供了海报和小册子，主修计算的 10 大理由，各计算领域介绍，FAQ 专栏，还有一些职业相关网站的链接。此外，美国劳工统计局（Bureau of Labor Statistics）还发表了美国数百个不同行业的职业展望，包括教育和培训需求、收入情况和就业前景。其中，关于软件工程师和计算机科学家的数据，可以参考 http://www.bls.gov/oco/ocos303.htm 和 http://www.bls.gov/oco/ocos304.htm。

1.4　关于计算的误解

下面介绍几个常见的对计算领域的误解，这些误解导致了很多人对计算存有偏见，也将很多天才拒之门外。

- 误解："研究计算的人都是内向的、书呆子式的极客。"

  事实：20 世纪 40 年代程序员的形象（见图 1-3）早已被颠覆了。今天，计算特别是编程，已经成为一个高度协作的职业。一个程序员有时候也可以写出一个规模很大的程序，但如今的大多数软件系统，无论其规模还是复杂性，都远不止是一两个人所能创造和控制的。计算领域是一个拼智商的领域，这里需要具有各种不同兴趣爱好的天才。遗憾的是，沉闷的媒体形象还是影响了很多本来很适合搞计算的年轻人。
- 误解："所有与计算相关的工作都会外包。"

  事实："所有"显然是一种夸大的说法，但除此之外，相当比例的外包工作是由于本地程序员供不应求，而不光是出于成本考虑。不是所有工作都适合外包：本地的拥有丰富技能的信息从业者具有天然优势。
- 误解："计算就是在桌面电脑或笔记本电脑上写代码。"

  事实：如今，嵌入设备中微处理器的数量已经超过个人电脑。机器人、智能武器、未来汽车、工业机械、（智能）手机和数字助理等，都是可以编程的对象，需要计算技能来制造和生产。
- 误解：".com 泡沫破灭之后，计算的好日子就到头了。"

  事实：.com 泡沫破灭是 21 世纪最初几年经济危机之后的事。在此之后，搜索、视频共享、社交网络、新闻传播和移动计算，可谓遍地开花。计算机越来越小、越来越快，也越来越便宜。几乎所有公司都有自己的官方网站，而人们也希望他们能够提供在线服务，并且能够通过自己的移动设备来使用。

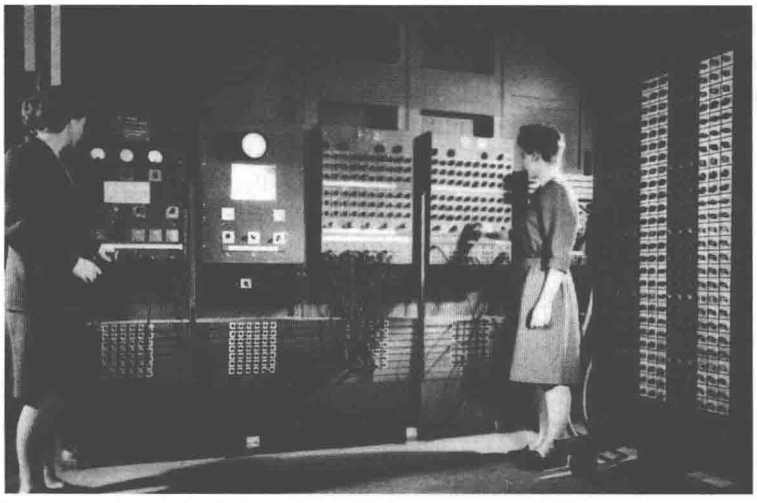


图 1-3 1947 年前后的程序员

- 误解："不必专门研究计算，艺术课或经济课老师会告诉我们怎么成为一个伟大的程序员。"
  事实：软件系统是迄今为止人类创造的最复杂的东西之一。通过构建这些系统得来的知识和经验，特别是从系统失败中得来的教训，都融汇到了计算机科学和软件工程的著作和训练中。经济和艺术怎么可能？

1.5 本章小结

- 计算机是一门自然科学，而不是人为的科学。信息处理的研究提供了生物系统、物质和能量、社会和经济系统中行为的模型。
- 计算领域的五大学科是计算机科学、软件工程、计算机工程、信息技术和信息系统。
- 计算人才适合很多行业，包括但不限于生物技术、航空航天、娱乐、信息检索、法律、商业、医药、媒体、游戏和网络安全。
- 很多关于计算的误解导致了这个领域人才（特别是年轻人才）的流失。

1.6 练习

本章的所有练习都需要读者做一些研究和搜索，仅阅读本章内容可能找不到答案。

1. 看一看 Peter Denning 的"Computing Is a Natural Science"[Den07]，然后写一篇读后感，两三段即可。
2. 找一些关于 DNA 的资料读一读。搞明白什么是"编码链"（coding strand）和"mRNA 合成"（mRNA synthesis）。这两个概念与你想象中的计算机程序有什么异同？
3. 本章前面列出的计算的五大学科，源自 ACM Computing Careers，网址是 http://computingcareers.acm.org/?page_id=6。这种分类十分宽泛，对编程感兴趣的人可能更多关注计算机科学和软件工程。请做一番研究，尝试给出计算机科学和软件工程的 20 个子学科。
4. 计算机科学还是一门新学科。搜集关于你现在就读的学校、毕业的学校，或者想去就读的学校的计算机科学老师的信息，看看他们本科的学位是哪个领域的。
5. 了解一下你中意的大学计算机科学系都开设了哪些跨学科的课程，列出个单子来。
6. 尝试寻找可靠的数据来回答下列两个问题：今天，世界上最流行的编程语言是什么？大一计算机科学课中最常讲授的编程语言是什么？

7. 如果你认识一些有编程经验的人，请做个访问调查，针对每个人分别记录，看他们最喜欢哪三种编程语言。再问问他们最不喜欢什么语言。随机应变：假如你的受访者非常激动，使用了刻薄的话来贬低某个语言，那么追问他们到底是该语言的设计者没有那个能力，还是受访者使用的语言与设计者当初的目的不相符。
8. 研究下列语言：JavaScript、Ruby、Io、Lua、Self、Java、Python、C、ActionScript、Smalltalk、LISP、Ada、bash、SQL 和 Go，找出与它们相关的一两个有趣的事实。
9. 自己动手做一个表 1-2 所示的电子表格。如果你从没用过电子表格软件，使用各种方法求助。
10. 如果你可以做出上一个练习要求的电子表格，那么插入过去（或将来）几个滑雪季的数据（表格行）。
11. 调查几种现在常用的加密方法，比如 Blowfish、AES 或 RSA。RSA 的应用范围有多广？为什么说 RSA 加密尚不知是否可证明安全（provably secure）？
12. 阅读或浏览与下列主题相关的文章各一篇：计算生物学、计算神经科学、计算社会科学、计算语言学。
13. 浏览 ACM TechNews：http://technews.acm.org/。不要只看当前这一周的文章，还要翻阅存档的文章（地址是：http://technews.acm.org/archives.cfm）。找出 10 到 20 篇能代表计算在当今世界应用范围的新闻报道来。
14. 关于人类的大脑相当于计算机的说法，有什么支持和反对的理由？
15. 读一读阿兰·图灵的传记。说出几个他被人们尊为当今计算机学科奠基者的原因。
16. 关于“谁是第一台机械计算机”这个问题的回答，可能要取决于人们对机械计算设备的定义。但对于第一台通用的、电子计算机倒是有共识的。这台计算机的名字叫什么？当初设计它的目的何在？谁是这台计算机的第一个程序员？
17. 今天，我们会自然而然地认为可以为任何设备编写程序。但原先可不是这样，早期计算机也可以编程，但是由人工使用计算机自己的机器语言编码指令。要想让为一台计算机编写的程序在另一台计算机上运行，必须把它编译（翻译）为另一台计算机的机器语言。Grace Murry Hopper 写的编译器可能是世界上最早的一个编译器。读一读海军上将 Hopper 的传记，看看她还为计算机科学做出过什么贡献？
18. 调查一下你中意的大学的商学院都开什么课。看看商学院会不会开设几何和微积分课程，或者要求学生必须去数学系选修这些课？看看商学院有没有编程课，或者要求学生必须去计算机系选修这类课程？
19. 哪种特许权能带来更多收入？《星球大战》电影的特许权，还是《疯狂橄榄球》的特许权？谁的利润更高？要把所有原始收入和延伸收入都计算在内。绝对准确的数据可能不容易获得，所以可以适当估算。注意在你的研究结论中给出引用的出处。
20. 找一篇严肃的研究论文，或者由权威机构发布的新闻报道，看看关于计算领域的负面宣传在多大程度上影响了人们进入这个领域（或在大学阶段没有选择这个方向）。媒体报道的准确性有多高？失真度又有多高？
21. 至少说出一个本章没有提到的社会公众对于编程或计算的误解。