

The
Pragmatic
Programmers

TURING

图灵程序设计丛书

通过解决问题锤炼软件开发技能

挑战编程技能

57道程序员功力测试题

[美] Brian P. Hogan 著

臧秀涛 译



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

挑战编程技能

57道程序员功力测试题

[美] Brian P. Hogan 著

臧秀涛 译

人民邮电出版社
北 京

图书在版编目 (C I P) 数据

挑战编程技能 : 57道程序员功力测试题 / (美) 布莱恩·霍根 (Brian P. Hogan) 著 ; 臧秀涛译. — 北京 : 人民邮电出版社, 2017. 2

(图灵程序设计丛书)

ISBN 978-7-115-44680-0

I. ①挑… II. ①布… ②臧… III. ①程序设计—习题集 IV. ①TP311.1-44

中国版本图书馆CIP数据核字(2017)第009877号

内 容 提 要

新手程序员在具备了理论基础后,面对实际项目时往往不知道如何解决问题;有经验的程序员在学习了一门新语言后,也会有很多不知道如何使用的特性。针对程序员的这一普遍困惑,资深软件工程师 Brian P. Hogan 在这本书中总结了 57 道练习题,帮助他们锤炼技能。这些练习题均取自实践,难度会逐渐增加,使得编程训练充满挑战又乐趣多多。

本书适合所有程序员阅读。

◆ 著 [美] Brian P. Hogan

译 臧秀涛

责任编辑 岳新欣

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>

北京隆昌伟业印刷有限公司印刷

◆ 开本: 880×1230 1/32

印张: 4.25

字数: 106千字

2017年2月第1版

印数: 1—3 500册

2017年2月北京第1次印刷

著作权合同登记号 图字: 01-2015-8282号

定价: 39.00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

Copyright © 2015 The Pragmatic Programmers, LLC. Original English language edition, entitled *Exercises for Programmers: 57 Challenges to Develop Your Coding Skills*.

Simplified Chinese-language edition copyright © 2017 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由 The Pragmatic Programmers, LLC.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

如何使用本书

宝剑锋从磨砺出。

音乐会钢琴家每天练习数小时，学习音乐，排练技艺，提升技能。他会反复练习同一段音乐，打磨每一处小细节，确保准确无误，因为等到登台表演时，面对那些花钱、花时间来的观众，他希望可以为自己的表现而感到骄傲。

职业足球运动员会在健身房里花费很多时间，练习托举、跑动和跳跃，并反复训练，直至完全掌握，然后才开始练习足球。他会研究比赛，观看以前的比赛录像。当然，他也会参加分组对抗和表演赛，确保做好一切准备，可以随时真正上场比赛。

空手道从业者一生都在练习“形”（kata）——这是模仿实战的动作套路，学习如何呼吸以及正确发力。他会千万遍地练习同一个套路，通过一次次重复做到越来越好。

我遇到过的优秀软件开发者，也是这样锤炼其技艺的。他们并不是天天只利用上班时间练习，而是会用自己的时间学习新编程语言，在其他方面不断精进自己的技术。当然，在工作中他们也会学习新东西，

但因为是工作时间，所以会有预期：公司是希望你有产出的，而不是花钱请你来练习的。

本书要谈的就是程序员如何锤炼其技能。翻到本书的某一页，打开文本编辑器，敲出上面的程序。可以自己做些修改。用你从来没用过的一门语言实现这个程序。随着一次次重复，你要做得越来越好。

目标读者

本书主要面向两类程序员。

首先，本书为刚开始学习编程的人提供了课外练习材料。技能不可能仅靠做作业就得到提升。未来的雇主会希望你们表现出批判性思维和解决问题的能力，而这需要通过练习来训练。本书会提供这种练习，书中的问题都取自实践，是很多开发者都要面对的，但是这里针对读者的能力做了针对性的设计。每一章都会围绕一项基本的编程能力展开，而且会比前一章复杂一点，后面的内容构建于前面所学的知识之上，同时让你为以后的挑战做好准备——不管是课内的还是课外的。

很多新手程序员习惯于有人清清楚楚地告诉他们如何解决某个问题。他们往往是跟着可以直接输入示例代码的教程来学习一门语言。这确实也是开始编写代码的不错途径。但是当面对没有现成答案的开放性问题时，这些程序员就纠结了。任何有经验的人都可以告诉你，软件开发中满是开放性问题。本书中的练习可以帮助你培养问题求解技能，使你有信心攻克更大的问题——或许是从没有人解决过的问题。

不过，本书也适合那些想将手头的事做得更好的有经验的程序员。在学习 Go 和 Elixir 时，我使用了类似本书中的那些程序。在尝试 iOS 开发时，我也编写了这些程序。每隔一段时间，我都会用一门我了解的语言实现一下它们。我熟练掌握了 JavaScript 和 Ruby，而看看是否能以不同的方式、不同的算法或模式实现其中的一个程序，也是极大的挑战。在我开始全职讲授 Ruby 和 JavaScript 时，这些程序帮我发现和解释了这两种语言的一些特性，这些特性我知道怎么用，但是并没有充分理解。所以，如果你是有经验的开发者，我鼓励你也这么做。比如用 Haskell 试试其中的一个程序；或是试试用你熟悉的每种语言编写其中的一个，然后比较一下结果。也可以让你的同事每周做其中的一个练习，并比较你们的方案；或是使用这些程序来指导团队中的新人。

教育工作者需要注意

如果你是在高中或大学讲授编程入门课程，可能会发现本书中的练习在课堂上也很有用。不过，我不建议将其用作最终的考试评估，我鼓励读者彼此分享其解决方案。当然，如果学生可以共同合作，我也建议将本书中的练习用于课堂。这些练习很适合问题导向的学习环境。

本书内容

之所以写作本书，首先是希望向编程语言初学者提供一些一开始可能要面对的、有挑战的问题。因此，大部分问题一开始相对简单，难度会逐渐增加。书中练习的这种延续性，使得编程基础的训练充满挑战，

同时又有很多乐趣，还可以让我们更加快速地上手一门新语言。在第一部分，程序只是简单地接受一些输入，操作数据得到不同的输出，让你体验计算机程序如何处理输入和输出操作。这是新手第一周要编写的程序。

下一步就有些挑战了，要编写进行计算的程序。有些程序就像计算房屋面积那么简单，但是其他程序可能会涉及金融和医疗方面的计算，和我们可能在工作中碰到的类似。

然后，通过引入决策逻辑和重复逻辑，我们会增加程序的复杂性，而且会引入函数。

之后，你会发现有些问题需要使用像数组（array）或映射（map）这样的数据结构来解决。这些程序还需要你利用之前解决的其他问题。

当然，没有点儿文件输入输出，程序是不完整的。所以我们会练习从文件中读数据，处理，最后写回。

现代的程序往往要和外部服务交互，因此你会发现一些程序需要使用第三方 API 的数据。

最后，本书后面有几个比较大的程序，需要读者综合前面所学。

此外，每个练习都有一些在构建程序时必须遵循的约束，还有一些高于当前程序的挑战。如果你完全没有编程基础，可以跳过这些挑战，等技能提高之后再回来看看。如果你已经有些经验，并且感觉程序太过简单，则可以立刻试试。取决于所选择的编程语言，有些挑战可能非常困难。比如，如果使用 JavaScript 和 HTML 开发这些程序，那实现一个 GUI 版本会非常容易。如果使用 Java 开发，就要多做大量的

工作。不过对于这些挑战，如果你感觉合适，可以自由修改。

然而，本书不会提供这些程序的解决方案。如果你做了足够深入的思考，而且使用了所有可以调配的资源，应该能独立想出解决之道，而这正是本书主旨所在。

最后一件事：本书中没有那些低劣的面试题，也没有 FizzBuzz 之类的问题。你不需要翻转二叉树，也不需要编写快速排序算法（除非它作为某种解决方案的一部分而用到）。如果你想找这类内容，应该到别的地方看看。那类问题有价值，但至于为什么要做，我们并不清楚，所以实际做起来会更困难。这就使得人们感觉它们不是那么好接近，因而成了学习的障碍。

本书中的问题是比较简单而且容易涉及的现实问题，可以帮你练习使用代码解决问题。

读者所需

你需要的就是自己喜爱的开发环境，甚至可以是从来没用过的那种。本书没有限定于某种具体的编程语言。请选择一门语言，抓起这门语言的参考指南，深入进去。但是提醒一句，编程语言的选择会影响书中程序实现的难易。比如，如果选择 Python 或 Ruby 来配合学习本书，开发图形用户界面就没那么容易。如果选择在浏览器中使用 JavaScript，那么使用外部文件和 Web 服务就会比其他语言复杂。与面向对象语言相比，如果选择的是函数式语言，则问题的解决方案会有显著区别。不过这正是这些练习的真正价值所在；它们可以帮你学习一门语言，并理解它与你之前学过的语言有何差别。

你应该能接入互联网，这样就可以尝试使用第三方服务的程序，并参与到本书的社区中来。

在线资源

本书的配套网站^①有一个论坛，你可以在那里和其他开发者探讨本书。你可以随意用自己钟爱的编程语言提交解决方案，并和其他读者探讨自己的方案。人们可以以不同方式解决问题，每个开发者都可以有自己的风格，这是编程最迷人的地方之一。

电子书

扫描如下二维码，即可购买本书电子版。



^① <http://pragprog.com/titles/bhwb>

致 谢

首先，感谢读者选择这本书，并愿意提升自己的软件开发技能。你真棒。这本书就是为像你这样的读者编写的，感谢阅读。

其次，感谢 Dave Thomas。感谢你相信本书的理念，也感谢你这些年来对我的指导。能向你学习是一种荣耀，我深感荣幸。在本书的写作上，你对我的鼓励意义非凡，感谢你投入很多时间审核书中的习题并提出建议。你和 Andy 让程序员的世界越来越好，非常感谢你们让我有机会以自己的方式作出一点小贡献。

特别感谢 Susannah Pfalzer。你让我的书变得更好。你似乎总能准确地抓住所有细节，而且指导我专注在真正重要的地方。这是你帮我做的第 6 本书，因为你这些年的指导，我才成了更好的作者。

还要感谢 Andy Hunt、Mike Reilly、Michael Swaine、Fahmida Rashid 和 Bruce Tate，感谢你们在我提出这些理念时给予我的鼓励。

本书中的程序都是我过去 10 年在编程教学中用过的。感谢 Zachary Baxter、Jordan Berg、Luke Chase、Dee Dee Dale、Jacob Donahoe、Alex Eckblad、Arrio Farugie、Emily Mikl、Aaron Miller、Eric Mohr、

Zachary Solofra、Darren Sopiartz、Ashley Stevens、Miah Thalacker、Andrew Walley，以及这些年上过我的课或者参加过我的培训的其他所有学生。你们对教学方法的反馈让我获益良多。感谢 Kyle Loewenhagen、Jon Cooley 和 George Andrews，你们的反馈和意见帮助我成长为一名更优秀的教师。

感谢 Deb Walsh，感谢你的鼓励，以及关于如何让学生学得更好的奇思妙想。关于教学，我们有共同的核心理念，在和你的交流中我也学到很多。感谢你与我分享自己的经验和专业知识，也感谢你对我的教学方法的支持。

本书凝结了很多新老开发者的很棒的反馈，这使得本书的习题衔接得更为合理、顺畅。出版前期审校人员花了很多时间和精力，用他们喜欢的编程语言实现书中的习题，并帮我找出了一些没有意义或需要改进的地方。感谢 Chris C.、Alex Henry、Jessica Janiuk、Chris Johnson、Aaron Kalair、Sean Lindsay、Matthew Oldham、Stephen Orr、Jason Pike、Jessica Stodola、Andrew Vahey 和 Mitchell Volk，感谢你们付出宝贵的时间测试这些习题，并提供建议和反馈。

感谢我的合作伙伴 Mitch Bullard、Kevin Gisi、Chris Johnson、Jeff Holland、Erich Tesky、Myles Steinhauser、Chris Warren 和 Mike Weber 对我的支持。

感谢我的妻子、最好的朋友 Carissa。你的爱和支持使本书成为可能。我会永远感激你为我以及女儿们所做的一切。

最后，感谢棒棒的 Ana 和 Lisa，感谢你们在我写作时给我拥抱，给我发信息。感谢你们在我写下这段文字时陪伴在我身旁。

目 录

第 1 章	将问题转变成代码	1
	理解问题	1
	发现输入、处理和输出	3
	用测试驱动设计	4
	用伪代码编写算法	7
	编写代码	9
	挑战	9
	前进!	10
第 2 章	输入、处理和输出	11
	1 问好	12
	2 计算字符数	13
	3 打印引语	14
	4 疯狂填词	15
	5 简单的数学处理	16
	6 计算退休时间	17
	本章回顾	17
第 3 章	计算	19
	7 矩形房间的面积	22

2 挑战编程技能：57道程序员功力测试题

8 比萨聚会	23
9 涂料计算程序	24
10 自助结账	25
11 货币兑换	26
12 计算单利	28
13 确定复利	30
本章回顾	31

第4章 作出决策

14 税额计算程序	36
15 密码验证	38
16 法定驾驶年龄	39
17 计算血液中的酒精含量	41
18 温度转换程序	43
19 计算身高体重指数	45
20 多州税收计算程序	47
21 从数字到名字	49
22 比较数字	50
23 定位汽车问题	51
本章回顾	52

第5章 函数

24 字母易位词检查程序	55
25 检查密码强度	56
26 计算还清信用卡欠款所需的时间	57
27 验证输入	59
本章回顾	60

第6章 重复

28 数字相加	65
---------------	----

29	处理错误的输入	66
30	乘法表	68
31	卡蒙内心率	69
32	猜数字游戏	71
	本章回顾	72

第 7 章 数据结构 73

33	神奇 8 号球	76
34	从员工列表中删除元素	77
35	选择优胜者	79
36	计算统计信息	81
37	密码生成器	83
38	过滤值	84
39	排序记录	85
40	过滤记录	87
	本章回顾	88

第 8 章 使用文件 89

41	姓名排序程序	91
42	解析数据文件	93
43	网站生成器	95
44	产品搜索	96
45	单词查找	98
46	词频统计	99
	本章回顾	100

第 9 章 使用外部服务 101

47	谁在太空中?	103
48	抓取天气	104
49	Flickr 照片搜索	105

4 挑战编程技能：57道程序员功力测试题

50 电影推荐 107

51 向 Firebase 提交笔记 109

52 创建自己的时间服务 110

本章回顾 111

第 10 章 完整的程序..... 112

53 待完成事项清单 113

54 短网址服务 114

55 文本分享 115

56 记录财产 116

57 多选琐事问答应用 117

下一步干什么？ 117

将问题转变成代码

如果你才接触编程，可能想知道有经验的开发者是如何看问题以及如何将其转变为可运行的代码的。其实，编写代码只是这个过程中很小的一部分。在解决问题之前，首先要将其分解。如果你观察过有经验的开发者，可能会觉得他们看上去就是打开代码编辑器，噼里啪啦敲出了一个解决方案。但是多年以来，他们分解的问题就算没有几千个，也有几百个了，他们能看出一些模式。而新手可能并不知道怎么做。所以，本章将研究一种分解问题并将其转变为代码的方法。你可以使用该方法解决本书其余部分的问题。

理解问题

要想知道应该做什么，最好的一个方法是把它写下来。如果我告诉你，我想要一个小费计算程序，是不是凭这个信息你就能开工写一个了？可能还不够。你可能还必须向我提一些问题。这通常称作收集需求，不过我喜欢把它看作找出程序应该提供的特性。

先考虑一些可以向我提出的问题，以便通过这些问题更好地理解我想要的是什麼。要构建这个应用，你需要了解什么呢？