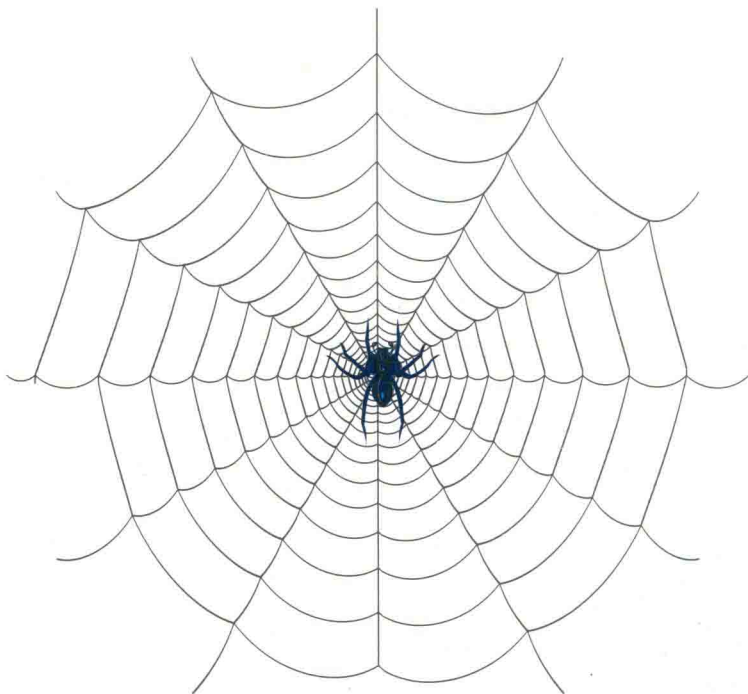


Exercises in Programming Style

编程风格

好代码的逻辑

[美] Cristina Videira Lopes◎著 顾中磊◎译



Python语言阐述

与**算法**和**数据结构**同等重要的程序设计概念了解编程和系统设计的不同方式，找寻**卓越代码**的奥秘，体会**编程之美**

TURING

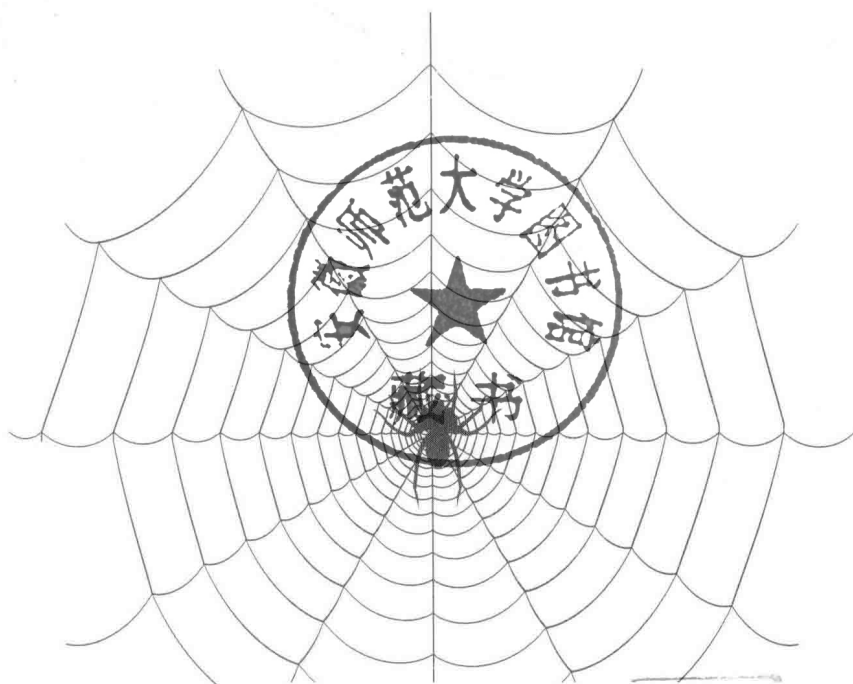
图灵程序设计丛书

Exercises in Programming Style

编程风格

好代码的逻辑

[美] Cristina Videira Lopes◎著 顾中磊◎译



人民邮电出版社

北京

图书在版编目 (C I P) 数据

编程风格 : 好代码的逻辑 / (美) 克里斯蒂娜·维
代拉·洛佩斯著 ; 顾中磊 译. — 北京 : 人民邮电出
版社, 2017.8

(图灵程序设计丛书)

ISBN 978-7-115-46035-6

I. ①编… II. ①克… ②顾… III. ①程序设计

IV. ①TP311.1

中国版本图书馆CIP数据核字(2017)第159088号

内 容 提 要

“同样的故事，不同的说法”，编程亦可如此。本书分别使用 33 种方法，在不同的约束定义下，实现了同一个词频统计任务，展示出风格迥异的编程风格，帮助读者加深对程序设计和系统设计方式的理解，为读者提供崭新的编程思路。

本书适合计算机专业的学生、有一定编程基础的程序员以及任何有志提高编程水平的人。

◆ 著 [美] Cristina Videira Lopes

译 顾中磊

责任编辑 朱 巍

执行编辑 夏静文

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: 800×1000 1/16

印张: 13.75

字数: 325千字

2017年8月第1版

印数: 1-3 500册

2017年8月河北第1次印刷

著作权合同登记号 图字: 01-2016-1574号

定价: 55.00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: 京东工商广登字 20170147 号

作者

Cristina Videira Lopes

美国加州大学欧文分校信息学专业教授，施乐帕罗奥多研究中心面向方向编程及AspectJ编程语言团队初创成员。获得过多项美国国家自然科学基金的资助，是全球唯一同时获得ACM杰出科学家与Ohloh Kudos 9级会员的人。

译者

顾中磊

1987年生，毕业于上海交通大学联读班。洛酷科技创始人。全栈开发工程师。自中学时代开始接触计算机编程，所参与项目涵盖各行各业，见证了技术革新对软件开发的改变。热衷于程序设计语言、软件架构设计与机器学习。



微信连接



回复“修炼”查看相关书单



微博连接

关注@图灵教育 每日分享IT好书



QQ连接

图灵读者官方群I: 218139230

图灵读者官方群II: 164939616

图灵社区
iTuring.cn

在线出版, 电子书, 《码农》杂志, 图灵访谈

版 权 声 明

Exercises in Programming Style 1st Edition / by Cristina Videira Lopes / ISBN:978-1-482-22737-6.

© 2014 by CRC Press.

Authorized translation from English language edition published by CRC Press, an imprint of Taylor & Francis Group LLC; All rights reserved. 本书原版由 Taylor & Francis 出版集团旗下 CRC 出版公司出版，并经其授权翻译出版。版权所有，侵权必究。

Post & Telecom Press is authorized to publish and distribute exclusively the Chinese (Simplified Characters) language edition. This edition is authorized for sale throughout Mainland of China. No part of the publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher. 本书中文简体翻译版授权由人民邮电出版社独家出版并限在中国大陆地区销售。未经出版者书面许可，不得以任何方式复制或发行本书的任何部分。

Copies of this book sold without a Taylor & Francis sticker on the cover are unauthorized and illegal. 本书封面贴有 Taylor & Francis 公司防伪标签，无标签者不得销售。

谨以此书献给 Julia。

前言

代码

本书的随书代码可以在以下网址下载：<http://github.com/crista/exercises-in-programming-style>。

目标读者

随书代码是本书的根基，这些代码面向每一个沉浸于编程艺术的人。我写这本书是为了补充和解释这些代码，因为它们的含义并非不言自明。资深的软件开发者能够重温其惯用的代码风格，也能了解其并不熟悉的别样曲风。

本书可以作为计算机科学和软件工程专业高级编程课程的教材，并附有讲义等其他教学资源。本书不适合作为编程入门课程的教材。正所谓“欲速则不达”，按部就班地学习尤为重要。通过学习，才能由“只有一种解法”向“一题多解”转变。本书的主要目标读者为大三大四的学生，或是低年级研究生。每章末尾的练习对读者的理解程度进行了测试。推荐阅读的内容更适合研究生。

本书同样适合写作者，尤其是那些有一些编程基础，或对编程技术有强烈兴趣的写作者。虽然存在巨大差别，但写作与编程也有许多共同点。

写作动机

20世纪40年代，法国作家雷蒙·格诺的著作*Exercises in Style*将同样一个故事以99种不同的写作风格描绘出来。该书是写作技巧的巅峰之作，很好地诠释了“同样的故事，不同的说法”。虽然故事本身微不足道且完全一样，但该书的亮点在于形式，而非内容。它说明了讲故事的方法影响着听众对故事的理解。

格诺的故事简单说来只有三句话。

叙事者在S公车上注意到一个长脖子、戴帽子的男子，他正在与旁边的人争吵。两小时后，叙事者又在圣拉扎尔火车站看到了这名男子，和他同行的还有他的朋友。这位朋友正在建议他在大衣上多加一颗扣子。

就这样！之后格诺就用99种不同的方式把故事复述了99遍，例如用反叙法、隐喻法和非人类视角等。

多年来，我教过许多编程课程，我发现学生一般很难理解不同的编程和系统设计方式。他们最多学过两门编程语言，所以只理解这些语言所推崇的风格，而难以接受其他风格。这并非他们的过错。回顾编程语言的历史，由于缺少关于计算机编程风格的教材，开发者在积累丰富经验之前是很难意识到这一点的。即使积累了丰富的经验，编程风格对于开发者可谓“言而不畅，聆而难明”，并且伴随许多技术争论。因此，受到格诺的启发，我决定用这些年来我所接触过的不同编程风格，完成完全相同的计算任务，以此为编程风格正名。

那么，什么是风格？在格诺的乌力波圈子（Oulipo，来源于法语，意为“潜在文学工场”）中，风格只不过是“在约束条件下完成创作”的结果，通常会基于一些数学概念，如组合或漏字文。这些约束是为了创作一些从学术角度来说有趣的作品，而与故事本身无关。这种写作方式盛行一时，造就了许多文学作品。

在本书中，编程风格也正是在特定的约束下完成编程的结果。约束既可以来自于外部，也可以是自己强加的；既可以是环境的挑战，也可以是人为的限制；既可以源于过往经历和可测量的数据，也可以仅是个人的喜好。撇开来源，约束是风格之种。正是应用不同的约束，我们才写出了层出不穷、风格迥异的代码，它们做的事完全相同，但做事的方式完全不同。

优秀的程序员必须知道，编程风格与数据结构和算法是同等重要的，前者更注重人文，而后者更偏重计算。代码不仅仅是向计算机发布指令，更是向阅读代码的人传达信息。无论何种表达^①，表达方式决定并影响着表达的内容如何被听众所认知。资深程序员不仅需要写出正确、健壮的代码，更需要为了不同的目的，选择合适的方式来编程。

然而，一般来说，传授算法和数据结构比起解释编程风格间的微妙差异要简单得多。算法和数据结构方面的图书通常遵循以下流程：伪代码、解释和复杂度分析。编程图书往往分为两类：一类解释编程语言，另一类展示程序设计方式或架构模式。然而，从运用编程语言自身倡导的各种理念，到组合最终构成程序的各个元素，编程其实是一个连续的整体；语言和模式互相配合，把它们一分为二是错误的。在我看来，格诺的工作将约束条件作为解释表达风格的基础，正是编程世界中将许多重要的创造性工作统一在一起的完美模型。

应当声明的是，我并不是第一个将约束视为解释软件系统风格的统一标准的人。关于“架构风格”（architectural style）方面的研究在很久之前就开始使用这种方法。必须承认，风格源于约束（禁止某事，某物必须存在，限制某事等）这个概念乍一看并不好理解。毕竟，谁愿意在约束

^① 文学或编程。——译者注

下编程呢？直至偶遇格诺的杰作，我才真正理顺这个概念。

与格诺的故事相似，本书涉及的计算任务微不足道：给定一个文本文件，统计文本中的词频，并按词频从高到低输出单词列表。通常，我们称之为词频统计。本书共33章，每章用一种编程风格来完成此词频统计任务，共33种风格。与格诺不同的是，我将列出每种风格的限制，并解释示例程序。我认为针对本书的目标读者，明确地点明会比读者自己理解更重要。每章会首先描述风格限制，然后给出示例程序，再详细解释代码。大部分章节会额外涉及在系统设计中如何使用该风格，以及该风格产生的前因后果。历史是重要的，任何一门学科都应不忘初心，我也希望读者会对拓展阅读有兴趣。

为什么是33种风格？因为个人能力有限。格诺的书有99种写作风格，如果我最初也以99种编程风格为目标，我想我可能永远无法完成这本书。然而，作为本书的基础，公开的GitHub代码仓库仍会不断地扩充。33种风格被分为9种类别：悠久历史、基本风格、函数组合、对象与对象交互、反射与元编程、异常处理、以数据为中心、并发和交互。整本书以这种分类方式进行组织，将相似的风格归类到一起。当然，应当也有其他的归类方式。

与格诺的作品相似，这些风格演练仅仅是“练习”。它们只是软件领域的草稿或琶音练习，并不是完整的乐章。真实的软件常常会在系统的不同部分应用多种不同的风格，并且所有的风格可以混合、配对、杂交。

最后，尽管格诺的作品是本书的灵感来源，但是软件与语言艺术并不完全相同。软件设计决策对应着效用函数；换言之，为了特定目的，一些编写方式要好于其他方式。^①在本书中，除了一些非常明确的情况，我尽可能避免给出好与坏的判断。因为好与坏不由我来评判，而是与每个项目的具体情况有关。

致谢

感谢Richard Gabriel、Andrew Black、Guy Steele、James Noble、Paul Stickler、Pual McJones、Lauri Tratt、Tijs van der Storm，以及加州大学欧文分校INF 212 / CS 235课程（2014年冬季）的学生，尤其是Matias Giorgio和David Dinh，你们为本书的初稿提出了有价值的反馈。

感谢IFIP 2.16工作组的成员。在我第一次表述本书想法时，他们的反馈对于本书成型至关重要。

感谢Peter Norvig、Kyle Kingsbury、Sara Triplet、Jorgen Edelbo、Darius Bacon、Eugenia Grabriellova、Kun Hu、Bruce Adams、Krishnan Raman、Matias Giorgio、David Foster、Chad Whitacre、Jeremy Mac Cabe和Mircea Lungu对于GitHub代码仓库的贡献。

^① 也许语言艺术也是如此，但我不太了解。

引 言

词频统计

如同格诺的故事，本书涉及的计算任务很简单：给定一个文本文件，显示出现频率最高的 N （如25）个单词，并且按词频由高到低进行排序。我们将大写字母转换成相应的小写字母，并且忽略the、for等停止词。简单起见，对出现次数相同的词的排序不作任何规定。通常，我们称这项任务为词频统计任务。

看一个输入输出样例：

输入：

```
White tigers live mostly in India  
Wild lions live mostly in Africa
```

输出：

```
live - 2  
mostly - 2  
africa - 1  
india - 1  
lions - 1  
tigers - 1  
white - 1  
wild - 1
```

如果对古登堡数字电子书计划中的简·奥斯汀的《傲慢与偏见》做同样的词频统计，会得到如下输出：

```
mr - 786  
elizabeth - 635  
very - 488  
darcy - 418  
such - 395  
mrs - 343  
much - 329  
more - 327  
bennet - 323
```

```
bingley - 306
jane - 295
miss - 283
one - 275
know - 239
before - 229
herself - 227
though - 226
well - 224
never - 220
sister - 218
soon - 216
think - 211
now - 209
time - 203
good - 201
```

本书将介绍这项词频统计任务。另外，每一章都会有练习，其中的一个练习是用该章所描述的风格编写其他简单的任务。以下是一些建议。

这些任务对于任何一名高年级学生都是非常简单的，考量的重点在于遵循每种风格中的限制，而非算法。

单词索引

给定一个文本文件，按字母顺序输出所有单词，连同每个单词出现的页码。忽略所有出现100次以上的单词。假定每45行为一页。以《傲慢与偏见》为例，前几行输出应当为：

```
abatement - 89
abhorrence - 101, 145, 152, 241, 274, 281
abhorrent - 253
abide - 158, 292
...
```

单词上下文

给定一个文本文件，对于给定的单词列表，按字母顺序输出单词及其上下文，连同出现的页码。假定每45行为一页，上下文为单词出现处前两个单词及后两个单词，忽略标点符号。以《傲慢与偏见》为例，concealment和hurt相对应的输出为：

```
perhaps this concealment this disguise - 150
purpose of concealment for no - 207
pride was hurt he suffered - 87
must be hurt by such - 95
and are hurt if i - 103
pride been hurt by my - 145
must be hurt by such - 157
infamy was hurt and distressed - 248
```

建议在此项任务中使用如下单词：concealment、discontented、hurt、agitation、mortifying、reproach、unexpected、indignation、mistake和confusion。

Python

本书的示例程序均用Python编写，但理解风格并不需要精通Python。事实上，每章的练习都包括用其他语言重写示例程序。因此，读者只需要能够读Python，而非写Python。

Python的代码很容易阅读，但语言中的若干特例可能会让使用其他语言的读者感到困惑。我在这里做一下简单介绍。

- **列表**。列表是使用特定语法的基础数据结构，和类C语言中的数组类似，例如`mylist = [0, 1, 2, 3, 4, 5]`。Python并没有数组类型^①，大多数在类C语言中使用数组的场景，在Python中会使用列表。
- **元组**。元组是不可修改的列表。元组同样也是使用特定语法的基础数据结构，和类Lisp语言中的列表类似，例如`mytuple = (0, 1, 2, 3, 4)`。列表和元组的处理方式相似，只是元组不可修改，因此所有对列表有修改的操作对元组并不适用。
- **列表下标**。列表和元组的元素可以通过下标来访问：`mylist[some_index]`。列表的下标下限和类C语言一样，为0。列表长度可以通过`len(mylist)`得到。列表的索引方式比这个简单的例子更丰富，以下是一些例子。
 - `mylist[0]`：列表的第一个元素
 - `mylist[-1]`：列表的最后一个元素
 - `mylist[-2]`：列表的倒数第二个元素
 - `mylist[1:]`：第二个元素至最后的子列表
 - `mylist[1:3]`：第二个元素至第四个元素的子列表
 - `mylist[::2]`：包含所有偶数下标元素的子列表
 - `mylist[start:stop:step]`：从`start`下标开始，在`[start, stop]`下标之间，每隔`step`个元素取一个而得到的子列表
- **上下界**。获取在列表长度之外的下标元素会导致下标错误。例如，在一个长度为3的列表中获取第4个元素（如：`[10, 20, 30][3]`）会导致下标错误。然而，许多针对列表（和集合）的Python操作在索引方面会做额外处理。例如，在一个长度为3的列表中获取第4个至第101个元素（如：`[10, 20, 30][3:100]`）会返回一个空列表（`[]`），而不是下标错误。类似地，当所需区间部分覆盖了列表的取值范围，Python会返回包含覆盖部分元素的列表，而不会返回任何错误（如：`[10, 20, 30][2:10]`会返回`[30]`）。这种自适应的处理方式可能会让用其他没有这种特性的语言编程的人非常困惑。

^① Python中有数组类型，但不是语言原生的数据类型，且没有特殊的语法。不如列表常用。

- 字典。在Python中，字典（或称映射）也是使用特定语法的基础数据结构，例如`mydict = {'a': 1, 'b': 2}`。这个字典将两个字符串映射成两个整型。通常来说，键与值可以是任何数据类型。在Java中，这种数据类型可以在HashMap类中找到；在C++中，表达形式会是map模板。
- self。在大部分面向对象语言中，一般都会有特殊的语法来表达对象对于自己的引用，如Java和C++中的this、PHP中的\$this以及Ruby中的@。与这些语言不同的是，Python没有这种特殊语法。同时，实例方法的第一个参数为一个对象，通常会被称为self，但并不是强制的，如以下例子：

```
1 class Example:
2     def set_name(self, n):
3         self._name = n
4     def say_my_name(self):
5         print self._name
```

两个成员方法的第一个参数均为self，都在方法内被引用。self这个词本身并不特殊，可以使用其他任何名字，如me、my甚至this，但任何self以外的词都会为Python程序员所反感。而在调用成员方法时，会有些许令人意外，因为第一个参数被省略了：

```
e = Example()
e.set_my_name("Heisenberg")
e.say_my_name()
```

这种参数个数上的不匹配，是因为点(.)在Python中是一种简单的语法糖(syntactic sugar)，更原始的写法应当是：

```
e = Example()
Example.set_my_name(e, "Heisenberg")
Example.say_my_name(e)
```

- 构造函数。在Python中，构造函数是名为__init__（init两边各有两条下划线）的普通函数。在对象实例被创建之后，这个函数会被自动调用，如以下例子：

```
1 class Example:
2     # 这是本类的构造函数
3     def __init__(self, n):
4         self._name = n
5     def say_my_name(self):
6         print self._name
7
8 e = Example("Heisenberg")
9 e.say_my_name()
```

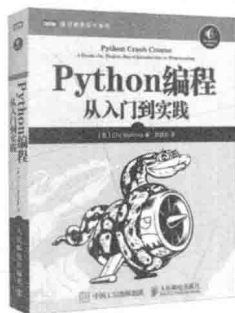
延 展 阅 读



- 你一定能看懂的算法基础书
- 代码示例基于Python
- 400多个示意图，生动介绍算法执行过程
- 展示不同算法在性能方面的优缺点
- 教会你用常见算法解决每天面临的实际编程问题

书号：978-7-115-44763-0

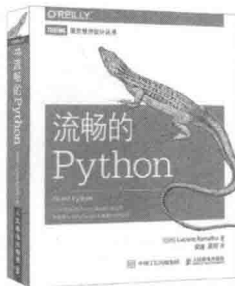
定价：49.00 元



- Amazon编程入门类榜首图书
- 从基本概念到完整项目开发，帮助零基础读者迅速掌握Python编程
- 上到有编程基础的程序员，下到10岁少年，想入门Python并达到可以开发实际项目的水平，本书是最佳选择！

书号：978-7-115-42802-8

定价：89.00 元



- PSF研究员、知名PyCon演讲者心血之作，Python核心开发人员担纲技术审校
- 全面深入，对Python语言关键特性剖析到位
- 大量详尽代码示例，并附有主题相关高质量参考文献和视频链接
- 兼顾Python 3和Python 2

书号：978-7-115-45415-7

定价：139.00 元

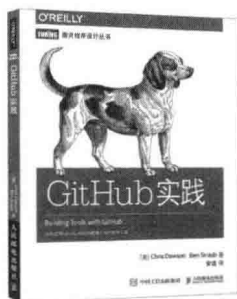


- 通过解决问题锤炼软件开发技能
- 练习均提炼于日常软件开发中经常遇到的实际问题
- 内容由浅入深，从简单的程序入手，逐步过渡到更难的问题

书号：978-7-115-44680-0

定价：39.00 元

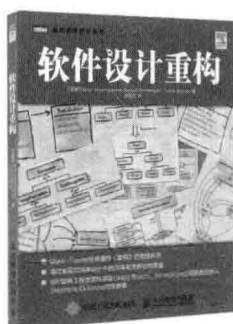
延 展 阅 读



- 满足独特的开发需求
- 活学活用GitHub，轻松构建属于你的软件工具
- 重于实践，介绍使用GitHub各项技术时应采取的折中方案以及注意事项

书号：978-7-115-43806-5

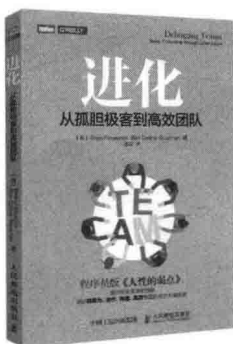
定价：69.00 元



- Martin Fowler经典著作《重构》的极佳补充
- 通过发现和消除设计中的坏味来改善软件质量
- IBM软件工程首席科学家Grady Booch、Synctique公司联合创始人Stéphane Ducasse作序推荐

书号：978-7-115-43124-0

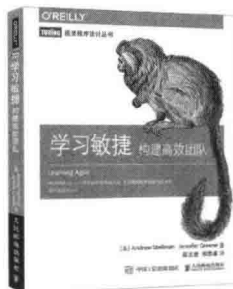
定价：59.00 元



- 技术人员版《人性的弱点》
- 提升职业生涯软技能
- 探讨领导力、合作、沟通、高效等团队成功关键因素
- 2016年最受欢迎电子书 非技术类TOP10

书号：978-7-115-43418-0

定价：45.00 元



- 精讲精益、Scrum、极限编程和看板方法
- 全面解读敏捷价值观及原则
- 提高团队战斗力

书号：978-7-115-44755-5

定价：79.00 元

目 录

第一部分 悠久历史

第 1 章 往日的美好..... 2

- 1.1 约束 2
- 1.2 代码 3
- 1.3 注解 5
- 1.4 系统设计中的应用 6
- 1.5 发展历程 7
- 1.6 拓展阅读 7
- 1.7 词汇表 8
- 1.8 练习 8

第 2 章 Forth 风格 9

- 2.1 约束 9
- 2.2 代码 10
- 2.3 注解 12
- 2.4 发展历程 14
- 2.5 拓展阅读 14
- 2.6 词汇表 14
- 2.7 练习 15

第二部分 基本风格

第 3 章 单片风格 18

- 3.1 约束 18
- 3.2 代码 19
- 3.3 注解 20
- 3.4 系统设计中的应用 21
- 3.5 拓展阅读 22

- 3.6 词汇表 22

- 3.7 练习 22

第 4 章 食谱风格 23

- 4.1 约束 23
- 4.2 代码 24
- 4.3 注解 25
- 4.4 系统设计中的应用 26
- 4.5 发展历程 27
- 4.6 拓展阅读 27
- 4.7 词汇表 27
- 4.8 练习 28

第 5 章 流水线风格 29

- 5.1 约束 29
- 5.2 代码 30
- 5.3 注解 31
- 5.4 系统设计中的应用 33
- 5.5 发展历程 33
- 5.6 拓展阅读 34
- 5.7 词汇表 35
- 5.8 练习 35

第 6 章 高尔夫风格 36

- 6.1 约束 36
- 6.2 代码 37
- 6.3 注解 37
- 6.4 系统设计中的应用 38
- 6.5 发展历程 39
- 6.6 拓展阅读 39