

原书第2版

程序设计导论

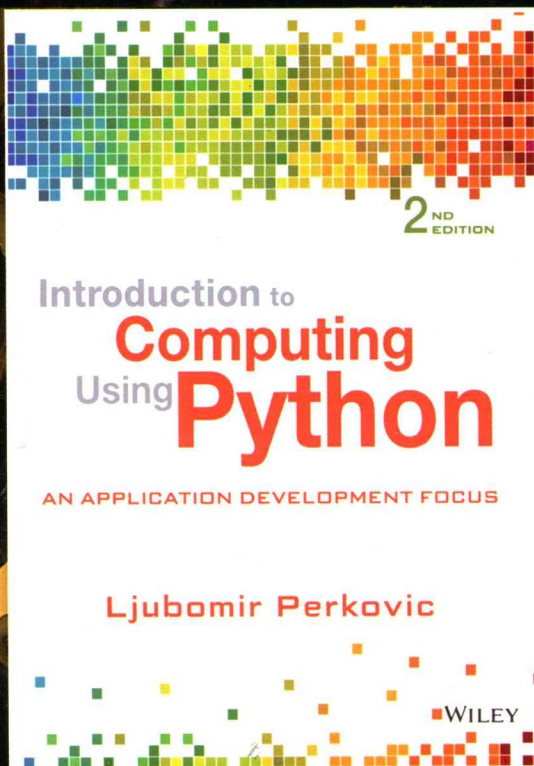
Python计算与应用开发实践

[美] 卢博米尔·佩尔科维奇 (Ljubomir Perkovic) 著

江红 余青松 译

Introduction to Computing Using Python

An Application Development Focus Second Edition



机械工业出版社
China Machine Press

算 机 科 学 丛 书

原书第2版

程序设计导论

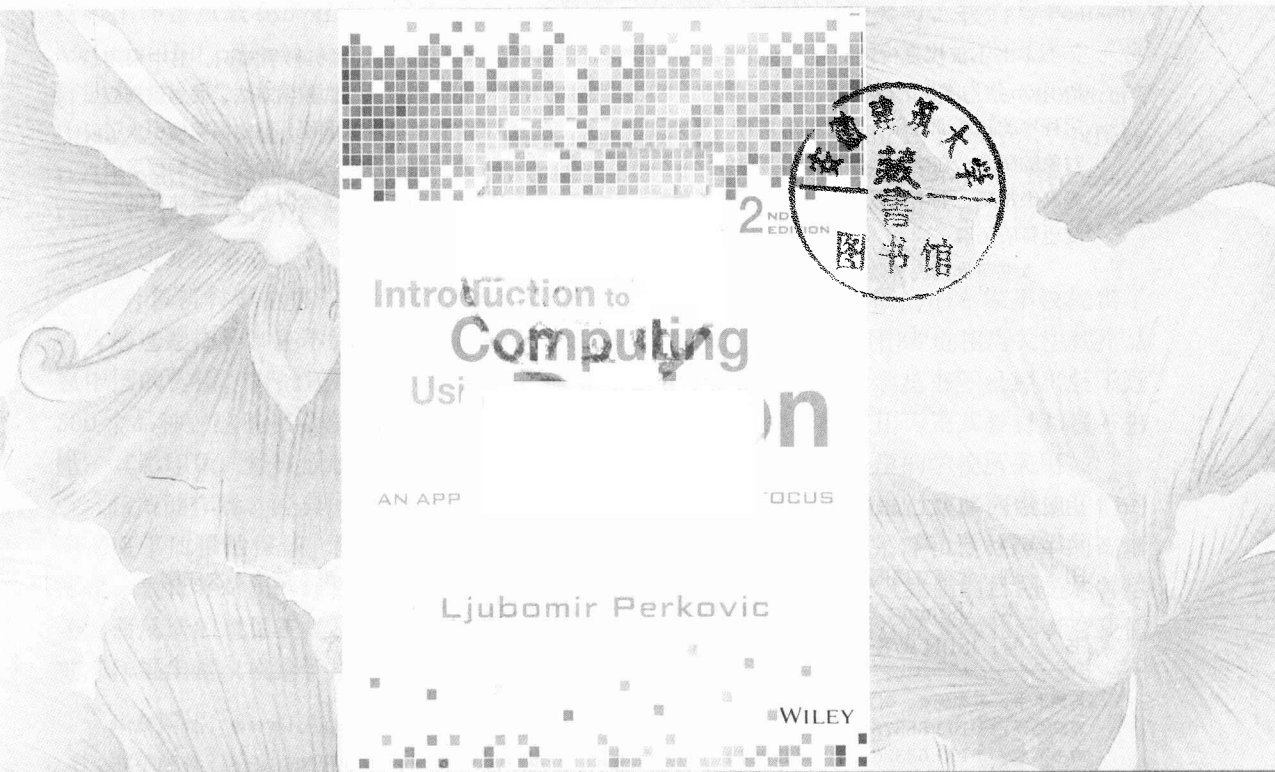
Python计算与应用开发实践

[美] 卢博米尔·佩尔科维奇 (Ljubomir Perkovic) 著

江红 余青松 译

Introduction to Computing Using Python

An Application Development Focus Second Edition



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

程序设计导论: Python 计算与应用开发实践 (原书第 2 版) / (美) 卢博米尔·佩尔科维奇 (Ljubomir Perkovic) 著; 江红, 余青松译. —北京: 机械工业出版社, 2018.10
(计算机科学丛书)

书名原文: Introduction to Computing Using Python: An Application Development Focus, Second Edition

ISBN 978-7-111-61160-8

I. 程… II. ①卢… ②江… ③余… III. 软件工具 - 程序设计 IV. TP311.561

中国版本图书馆 CIP 数据核字 (2018) 第 235232 号

本书版权登记号: 图字 01-2018-4065

Copyright © 2015 John Wiley & Sons, Inc. All rights reserved.

All rights reserved. This translation published under license. Authorized translation from the English language edition, entitled Introduction to Computing Using Python: An Application Development Focus, Second Edition, ISBN 978-1-118-89094-3, by Ljubomir Perkovic, Published by John Wiley & Sons. No part of this book may be reproduced in any form without the written permission of the original copyrights holder.

本书中文简体字版由约翰·威立父子公司授权机械工业出版社独家出版。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。

本书封底贴有 Wiley 防伪标签, 无标签者不得销售。

本书不仅仅是传统的程序设计导论性教材, 而且囊括了包罗万象的计算机科学知识。书中采用 Python 作为学生的第一门程序设计语言, 提出“正确的时刻 + 正确的工具”的教学方法, 尤为重视应用程序的开发训练和计算思维的培养, 易于教学和实践。书中首先讲解命令式编程, 但也很早便引入了对象的概念; 在学生具备足够的基础知识也更有学习动力之后, 开始讲解自定义类、面向对象编程等更多高级技巧。书中涵盖了问题求解的核心技术和各类经典算法, 这一版还增加了案例章节, 所涉及的广度也有所拓展。

本书可作为高等院校计算机科学和程序设计专业学生的教材或教学参考书。

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 卢璐

责任校对: 李秋荣

印刷: 北京文昌阁彩色印刷有限责任公司

版次: 2019 年 1 月第 1 版第 1 次印刷

开本: 185mm × 260mm 1/16

印张: 24.75

书号: ISBN 978-7-111-61160-8

定价: 99.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的优势，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅肇划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与Pearson、McGraw-Hill、Elsevier、MIT、John Wiley & Sons、Cengage等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Andrew S. Tanenbaum、Bjarne Stroustrup、Brian W. Kernighan、Dennis Ritchie、Jim Gray、Afred V. Aho、John E. Hopcroft、Jeffrey D. Ullman、Abraham Silberschatz、William Stallings、Donald E. Knuth、John L. Hennessy、Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力相助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专门为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近500个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

华章网站：www.hzbook.com

电子邮件：hzsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



华章科技图书出版中心

译者序

Introduction to Computing Using Python: An Application Development Focus, Second Edition

本书是一本基于 Python 应用程序开发实践的计算机程序设计导论课的教程，不仅可以作为程序设计的入门教程，更提供了计算机科学概念和现代计算机应用程序开发工具的广泛知识和应用。

本书采用面向问题的叙述方式，即在适当的时刻引入相关的计算概念、算法技术、Python 结构和其他工具，而不是逐一罗列计算的概念和 Python 语言结构知识。本书提供了大量基于 Python 交互式命令行的示例，鼓励学生动手实践。书中还包含大量的练习题、习题和思考题，可以进一步巩固和拓展读者学到的知识。本书还包括额外的 11 个案例研究（可访问华章网站 www.hzbook.com 下载），综合展示对应章节中所涉及的概念和工具，可以引导读者提高解决实际问题的能力。

本书以“广度优先”的方式组织内容，共分为四个部分：计算机科学导论和 Python 基础（第 1～3 章）、基于 Python 的算法设计和问题解决（第 4～6 章）、基于 Python 的复杂应用程序开发（第 7～9 章）、知识深入和高级应用（第 10～12 章）。本书由浅入深，理论知识和实际应用相结合，逐步引导读者学会使用计算机程序设计解决各种问题。

本书的另一大特色是讲解细致，正文中使用了大量的图和表等，使读者更容易阅读和理解正文内容。对于程序设计过程中可能出现的潜在陷阱，本书以“注意事项”的形式给出警示。书中还采用“知识拓展”的形式来简要地探索有趣但稍微偏离正文内容的主题。

本书是 DePaul（德保罗）大学的精品课教程，在其提供的教学官网（www.wiley.com/college/perkovic）中包含大量的教学辅助内容，无论是教师、助教、学生，还是一般读者，均可以从本书教学官网中获取与本书内容相关的额外信息的资源库。

本书由华东师范大学江红和余青松共同翻译。衷心感谢机械工业出版社华章公司的编辑曲熠积极帮我们筹划翻译事宜并认真审阅翻译稿件。在本书翻译的过程中我们力求忠于原著，但由于时间和学识所限，且本书涉及各个领域的专业知识，不足之处在所难免，敬请诸位同行、专家和读者指正。

江红 余青松

2018 年 8 月

本教程介绍程序设计、计算机应用程序开发和计算科学的基础知识及应用实践，适用于大学水平的程序设计导论课程。本教程不仅仅是程序设计的入门教程，更提供了计算机科学概念和现代计算机应用程序开发工具的广泛知识和应用。

本教程采用的计算机程序设计语言是 Python——一种比大多数语言学习曲线更加平滑的语言。Python 提供了强大的软件库，使得复杂的任务很容易上手，例如开发图形应用或者查找 Web 网页上的所有超链接。在这本教科书中，我们充分利用 Python 语言的易学性和易用性，同时使用 Python 库进行更多的计算机科学研究，并将重点放在现代应用程序开发上。这样做的好处是使得本书充分介绍了计算和现代应用程序开发领域的相关知识和应用。

本教程的教学方法是以广度优先的方式介绍计算的概念和 Python 程序设计知识。本教程的方法更接近自然语言的学习方法，从若干通用的词汇开始，逐渐扩展相关知识，而不是逐一罗列计算的概念和 Python 语言结构知识。本教程采用面向问题的叙述方式，只有在需要的时候才介绍相关的计算概念、Python 结构、算法技术和其他工具，即采用了“在正确的时间使用正确的工具”的模型。

本教程采用了命令式编程优先和面向过程的程序设计理念，但并不回避在早期讨论对象的概念。当激发了学生的兴趣并做好思想准备之后，再讨论用户自定义的类和面向对象的程序设计。教科书的最后三章和相关的案例研究使用 Web 爬取、搜索引擎和数据挖掘的上下文来介绍一系列广泛的主题。这些主题包括有关递归、正则表达式、深度优先搜索、数据压缩和谷歌的 MapReduce 框架的基本概念，以及诸如图形用户界面组件、HTML 解析器、SQL、JSON、输入/输出流以及多核编程的实用工具。

这本教科书适用于计算机专业计算机科学和程序设计知识的课堂教学。本教程覆盖了广泛而又基本的计算机主题以及当前流行的技术，这些有助于学生广泛理解该领域，并有信心开发与 Web 或数据库交互的“真正”的现代应用程序。教科书广泛的知识覆盖也使得它非常适合于那些同时需要掌握程序设计和计算概念但又不愿意选修一门或两门以上计算课程的学生。

技术特性

本教程具有许多吸引学生的特性，并积极鼓励学生动手实践。首先，本书提供了大量基于 Python 的交互式命令行的示例。学生可以很容易地自己复制这些代码行。通过运行这些代码并观测代码的执行结果，学生可能会使用交互式命令行的即时反馈来进行更深入的实验。

贯穿整本教程，我们将一些练习题穿插在正文当中，其目的是加深对正文中刚刚讨论过的概念的理解。这些问题的答案包含在相应的章节或者是案例研究的末尾，以允许学生检查他们的答案是否正确，或者在思路堵塞的情况下作为参考。

对于程序设计过程中可能出现的潜在陷阱，本教程以“注意事项”的形式来警示学生。

教程中还使用“知识拓展”的形式来简要地探索有趣但稍微偏离正文内容的主题。正文中大量的练习题、图和表等，为正文内容提供了更棒的视觉效果，从而使学生更容易阅读和理解正文内容。

最后，本教程各章结尾均包含了大量的思考题，其中许多思考题与入门级教科书中常见的思考题截然不同。

本教程的电子版还提供额外的教学材料，其中包括 11 个案例研究[Ⓒ]。每一个案例研究都与一章（第 2 ~ 12 章）的正文内容相关联，并充分展示对应章节中所涉及的概念和工具。案例研究中包括额外的思考题，以及相应的练习题及其答案。

在线补充资料[Ⓓ]

在本教程的配套网站上，提供了以下补充资料：

- 每个章节的 PowerPoint 教学幻灯片
- 每个章节的学习目标
- 教程中出现的所有代码示例
- 习题和思考题的参考答案（仅供教师使用）
- 考试题（仅供教师使用）

致学生：如何阅读本教程

本教程的目的是帮助读者掌握程序设计和开发计算思维的技能。程序设计和计算思维是实践行为，除了需要一台安装了 Python 集成开发环境的计算机以外，还需要用于演算的纸和笔。理想情况下，当读者阅读本教程的时候，必须拥有这些工具。

本教程大量使用了 Python 的交互式命令行示例。请读者尝试在命令行中运行这些示例。欢迎读者进一步实验。请读者放心，即使你不小心犯了错误，计算机也不大可能大发雷霆的！

读者还应该尝试完成正文中给出的所有练习题。练习题的参考答案位于相应章节的结尾。如果你思路堵塞了，去偷看一眼参考答案也可以，但是偷看一眼之后，请尝试自己解决问题而不要继续偷看。

对于编程过程中潜在的陷阱，在正文中使用“注意事项”的形式来警示读者。这些警示是非常重要的，读者阅读时不应该跳过。“知识拓展”部分则讨论与主题稍微相关的话题，读者愿意的话阅读时可以跳过，或者感兴趣的话也可以更加深入地探索这些话题。

在阅读正文的某些内容的时候，读者可能会灵感闪现，想开发自己的应用程序，也许是一个纸牌游戏，或一个实时跟踪一系列股票市场指数的应用程序。如果灵感闪现，那就勇敢地去尝试吧！相信你一定会收获满满。

本教程概述

本教程共分 12 章，以“广度优先”的方式介绍了计算概念以及 Python 程序设计语言。本教程的电子版还包括案例研究，展示了教程各章中所涵盖的概念和工具。

Ⓒ 关于电子教程案例研究内容，有需要的读者请到华章网站 (www.hzbook.com) 下载。——编辑注

Ⓓ 关于本书教辅资源，只有使用本书作为教材的教师才可以申请，需要的教师可向约翰·威立出版公司北京代表处申请，电话 010-84187869，电子邮件 sliang@wiley.com。——编辑注

Python 和计算机科学导览

第 1 章介绍基本的计算概念和术语。首先讨论计算机科学是什么以及开发人员做什么，并定义建模、算法设计和程序设计的概念。然后描述了计算机科学家和应用程序开发人员的工具包，从逻辑到系统，重点在于程序设计语言、Python 开发环境和计算思维。

第 2 章介绍核心的内置 Python 数据类型：整型、布尔型、浮点型、字符串、列表和元组。本章使用 Python 交互式命令行的方式阐述不同数据类型的特点。介绍没有侧重全面性，而是侧重每种数据类型的用途，以及数据类型之间的差异和相似之处。这种方法可以激发对对象和类的更抽象的讨论，而这对于最终掌握数据类型的正确用法是必需的。本教程电子版中的案例研究 (CS.2) 充分利用了这些讨论，从而引入了海龟图形类，让学生能够交互式地绘制简单有趣的图形。

第 3 章介绍命令式和面向过程的程序设计，包括基本的执行控制结构。本章将程序作为存储在文件中的 Python 语句序列。为了控制语句的执行方式，引入了基本条件和迭代控制结构：单分支和双分支 `if` 语句，以及迭代一个显式序列或数字范围的最简单的 `for` 循环模式。本章介绍了函数，作为一种封装小应用程序的方式；本章还在第 2 章所涵盖的对象和类的知识上，描述了 Python 如何赋值和传递参数。本教程电子版中的案例研究 (CS.3) 通过基于海龟图形的可视化上下文，激发读者通过程序实现自动化，并通过函数实现抽象。

前三章对 Python 程序设计和计算机科学提供了一个浅显而广泛的介绍。通过介绍 Python 的核心数据类型和基本执行控制结构，学生能够尽早上手编写简单而完整的程序。同时，在早期介绍函数可以帮助学生理解程序的功能，即程序所需要的输入是什么，以及程序产生的输出是什么。换言之，函数的抽象和封装是用来帮助学生更好地理解程序的。

专注于算法思考

第 4 章更深入地讨论了文本和文件处理。本章继续讨论第 2 章中涉及的字符串知识：字符串值的表示、字符串运算符和方法，以及格式化输出。文件输入/输出 (I/O) 也会介绍，特别是读取文本文件的不同模式。最后，使用文件 I/O 的上下文来激发对 Python 中异常和异常类型的讨论。本教程电子版中的案例研究 (CS.4) 讨论了图像文件（通常存储为二进制文件而不是文本文件）是如何读取和写入的，以及如何使用 Python 处理图像。

第 5 章深入介绍执行控制结构和循环模式。基本条件和迭代结构在第 3 章中介绍，然后在第 4 章中使用（例如，在读取文件的上下文中）。第 5 章一开始先讨论多分支条件语句，其余大部分篇幅则用于描述不同的循环模式：`for` 循环和 `while` 循环的各种不同使用方法。在讨论嵌套循环模式时，还引入了多维列表。本章作为核心章节，不仅涵盖了 Python 循环结构，还描述了问题分解的不同方式。因此，本章从本质上讨论了问题求解和算法。本教程电子版中的案例研究 (CS.5) 分析了图像处理的底层原理，描述了如何实现经典的图像处理算法。

第 6 章详细介绍了 Python 内置容器数据类型及其用法。引出字典、集合和元组数据类型加以介绍。本章还完成了对字符串的介绍，并讨论了字符编码和 Unicode。最后，在讨论选择和排列容器中的项时引入了随机性的概念。本教程电子版中的案例研究 (CS.6) 利用本章中介绍的概念，展示了如何开发一个 21 点扑克牌游戏应用程序。

第 4 ~ 6 章代表了本教程所采取的“广度优先”方法的第二个层次。在入门程序设计课程中，学生所面临的主要挑战之一是掌握条件和迭代结构，更一般地说，是掌握解决计算问题和设计算法的技能。关键的第 5 章（关于如何应用执行控制结构的模式）出现在学生

学习了基本条件语句和迭代模式的几个星期后，此时他们已经渐渐适应了 Python 语言。对 Python 语言和迭代有一定程度的熟悉之后，学生可以专注于算法问题，而不是那些诸如如何正确地读取输入或者格式化输出的次要问题。

管理程序的复杂性

第 7 章将重点转移到软件开发过程本身和管理更大、更复杂程序的问题上。本章介绍了名称空间。名称空间是管理程序复杂性的基础。本章建立在第 3 章函数和参数传递的基础上，引出了代码重用、模块化和封装的软件工程目标。函数、模块和类是可以用来实现这些目标的工具，本质上是因为它们定义了单独的名称空间。本章描述了如何在正常控制流和异常控制流（当异常由异常处理程序处理时）中管理名称空间。本教程电子版中的案例研究（CS.7）基于本章的内容展示了如何使用调试器查找程序中的错误，或者更一般地，如何使用调试器分析程序的执行情况。

第 8 章涵盖了 Python 中新类的开发和面向对象程序设计（OOP）的范式。本章以第 7 章揭示的“类通过名称空间实现”为基础，解释如何开发新的类。本章通过运算符重载（Python 设计理念的中心）介绍了面向对象程序设计的概念，以及继承（强大的面向对象程序设计属性，将在第 9 章和第 11 章加以应用）。通过抽象和封装，类实现了模块化和代码重用的软件工程目标。然后通过抽象和封装的讨论来引出用户自定义的异常类。本教程电子版中的案例研究（CS.8）进一步阐述了用户自定义容器类中迭代行为的实现。

第 9 章介绍了图形用户界面（GUI），展示了面向对象方法在开发图形用户界面中的强大之处。本章使用 Python 的 Tk 组件工具包，它是 Python 标准库的一部分。本章中讨论如何利用交互式组件实现事件驱动编程模式。除了介绍图形用户界面开发外，本章还展示了如何使用面向对象程序设计的强大功能来实现模块化和可重用的程序。本教程电子版中的案例研究（CS.9）通过实现基本计算器图形用户界面的过程证实了这一强大功能。

第 7 ~ 9 章的主要目标是向学生介绍程序复杂性和代码组织问题。这几章描述如何使用名称空间来实现功能的抽象和数据的抽象，并最终实现封装的、模块化的、可重用的代码。第 8 章全面讨论了用户自定义类和面向对象程序设计。然而，面向对象程序设计的优越性在实际应用中才能最好地体现，而这正是第 9 章的内容。其他有关面向对象程序设计的应用和实例将在后续章节陆续讨论，特别是 11.2 节、12.3 节、12.4 节以及第 10 章的案例研究 CS.10。第 7 ~ 9 章为学生将来在数据结构和软件工程方法方面的学习提供了基础。

知识深入和高级应用

第 10 ~ 12 章是本教程的最后三章，涵盖了各种高级主题，从基本的计算机科学概念（例如递归、正则表达式、数据压缩和深度优先搜索等）到实用的现代工具（例如 HTML 解析器、JSON、SQL 和多核编程等）。文中通过开发诸如 Web 爬虫程序、搜索引擎和数据挖掘应用程序来引出这些高级主题并将它们连接起来。然而，这些主题是松散的，并且每一个单独的主题都是独立呈现的，目的是允许教师根据他们认为合适的材料来设计不同的应用上下文和主题。

第 10 章介绍了计算机科学的基本主题：递归、查找和算法的运行时间分析。本章一开始即讨论递归思想。然后将这种技巧应用于从绘制分形图到病毒扫描的各种各样的问题上。本章最后一个例子用于阐述深度优先搜索。递归的优点和缺点导致算法运行时间分析的讨论。然后将算法运行时间的分析应用于各种查找算法性能的分析。本章把重点放在计算的理论方面，以便为今后的数据结构和算法课程奠定基础。本教程电子版中的案例研

究 (CS.10) 讨论了汉诺塔问题, 展示了如何开发一个可视化的应用程序来说明递归解决方案。

第 11 章介绍了万维网——一个中央计算平台, 同时也是一个创新计算机应用程序开发的巨大数据源。在讨论访问 Web 上的资源和解析 Web 页面的工具之前, 对 Web 语言 HTML 进行了简要讨论。为了从 Web 页面和其他文本内容中抓取所需的内容, 首先介绍了正则表达式。在入门课程中接触 HTML 解析和正则表达式的好处是, 学生在学习正规语言课程之前, 将熟悉其在应用中的用法。本教程电子版中的案例研究 (CS.11) 利用本章中所涉及的不同主题来展示一个基本的 Web 爬虫程序的开发过程。

第 12 章介绍数据库和大型数据集的处理。在讲述如何存储从网页中抓取的数据时, 简要地介绍了数据库查询语言 SQL 以及一个 Python 数据库应用编程接口。鉴于当今计算机应用数据库的普及, 建议学生尽早接触数据库及其使用 (如果没有其他的理由, 应该在第一次实习前熟悉数据库)。数据库和 SQL 的讨论只是介绍性的, 应该被看作以后数据库课程的基础。本章还讨论了如何利用计算机上可用的多个内核更快速地处理大数据集。本章还介绍了谷歌公司的问题解决框架 MapReduce, 并在此应用中介绍了列表解析和函数式编程范式。本章为进一步研究数据库、程序设计语言和数据挖掘奠定了基础。本教程电子版中的案例研究 (CS.12) 采用这一背景来讨论数据交换, 或者如何格式化并保存数据, 以便任何需要这些数据的程序可以方便高效地访问它们。

第 2 版新内容

本教程的第 1 版和第 2 版之间的最大变化是结构性调整。各章所涵盖的基本知识和用于描述基本概念的案例研究在第 2 版中实现了明确的分离。案例研究已经从各章节中分离出来, 在第 2 版中包含在教程的电子版本中。这种结构性变化有两个好处。第一个好处是, 教科书章节可以更加专注于基本知识。第二个好处是, 可以为案例研究提供更多的空间。新版本中出现了四个新的案例研究, 教程中每一章 (除了“非技术性”的介绍性章节) 都关联了一个案例研究。

除了这种结构性的变化, 教程还增加、删除了一些内容, 纠正了一些错误, 改进了一些表述方式。以下我们将一一罗列出这些变化。

在第 2 章中, 我们增加了元组数据类型的讨论 (包含在第 1 版的第 6 章中)。这一举措是合理的, 因为在 Python 中, 元组数据类型是一种关键的内置数据类型, 并被许多标准库模块和 Python 应用程序所使用。例如, 与第 4 章和第 5 章相关的案例研究中讨论的图像处理模块就使用了元组对象。因为元组数据类型与列表数据类型非常相似, 所以增加这个内容不会让第 2 章的讨论时间延长多少。

在第 3 章中, 改进了阐述函数的方式。特别是提供了更多的例子和练习题以帮助说明如何传递不同数量和类型的函数参数。第 4 章的案例研究被替换为新的关于处理图像文件的应用程序。新的案例研究给了学生一个令人兴奋的机会, 他们可以在视觉媒体的应用中查看教程内容。同时, 处理和格式化日期、时间字符串的内容被移动到 4.2 节。在第 2 版中, 重要的第 5 章有一个实现图像处理算法的相关案例研究。这部分内容再次利用视觉媒体的吸引人的应用过程来阐述基本概念 (例如嵌套循环)。

第 6 章不再包括元组数据类型的讨论 (被移至第 2 章中)。在第 2 版的第 7 章中包括了一个调试和调试器使用的相关案例研究。它有效地利用了本章所涵盖的概念, 为学生提供

了一种新工具，帮助他们进行程序的调试。第 8 章和第 9 章只是略有变化。第 10 章对线性递归及其与迭代的关系进行了更为深入的研究。第 11 章几乎没有变化。最后，在第 2 版的第 12 章中提供了一个数据交换的相关案例研究，它将帮助学生获得使用数据集的相关实践经验。

最后，第 2 版教程中增加了大约 60 道练习题和章节后面的思考题。

致教师：如何使用本教程

本教程的内容是为两个学期的课程设计的，主要针对计算机科学和计算机科学程序设计专业的学生。本教程的内容足够一个典型的 15 周的课程使用（可能正好适合于准备充分并且积极性很高的学生）。

本教程的前六章全面覆盖了 Python 语言中命令式 / 面向过程的程序设计部分。它们应该按顺序讨论，但也可以在学习第 4 章之前学习第 5 章。此外，还可以跳过第 6 章的内容，然后在需要的时候回过头来再学习。

为了有效地展示面向对象程序设计，建议按照顺序依次学习第 7 ~ 9 章的内容。在学习第 8 章之前建议先学习第 7 章，这点非常重要，因为第 7 章揭开了 Python 类实现的神秘面纱，从而使得学生可以更加有效地学习面向对象程序设计主题（例如运算符重载和继承）。同样，在学习第 8 章之后再学习第 9 章也是非常有益的（但不是必须如此），因为第 8 章提供了一个应用，其中展示了面向对象程序设计的巨大优越性。

第 9 ~ 12 章都是可选内容，它们仅仅依赖于第 1 ~ 6 章的内容（当然也有少量扩充的知识点），其包含的内容一般可以跳过或者由任课教师自由编排授课顺序。扩充的知识点位于 9.4 节（它演示了如何使用面向对象的程序设计方法开发图形用户界面），以及 11.2 节、12.3 节和 12.4 节（它们都使用了用户自定义的类）。所有这些知识均依赖于第 8 章中的内容。

在授课中使用本教程但计划将有关面向对象程序设计的知识留给后续课程的教师，可以先讲授第 1 ~ 7 章的内容，然后从第 9 ~ 12 章中选择非面向对象程序设计部分的主题内容进行授课。对于那些希望讲授面向对象程序设计知识的教师，应该使用第 1 ~ 9 章的内容，然后从第 10 ~ 12 章中选择相应主题进行授课。

致谢

本教程第 1 版的内容材料是在 DePaul 大学教授 CSC241/242 课程序列（计算机科学导论 I 和 II）的三年多时间里开发设计的。在这三年中，6 个不同年级的计算机科学专业的新生学完了本课程系列。我在不同的学生群体中尝试不同的教学方法，重新安排和重组教程中的内容材料，并尝试教授给学生入门级程序设计课程中通常不教的主题内容。不断的重组和实验使得课程内容材料不太流畅，但更具挑战性，特别是对于早期的学习群体。令人惊奇的是，虽然学生在本课程中所得到的分数不高，但他们依旧热情不减，这反过来又帮助我维持了热情。我衷心感谢他们。

我衷心感谢 DePaul 大学计算机学院的教师和管理人员，他们创造了一个真正独特的学术环境，鼓励教育实验与创新。他们中的一些人也直接参与了本教程的创作和修订。副院长 Lucia Dettori 合理安排了 my 的课程以便我有时间写作。Curt White 是一位经验丰富的教科书作者，他积极鼓励我开始写作，并极力向 John Wiley & Sons 出版社推荐我。Massimo

DiPierro 是 web2py Web 框架的创始者，同时也是我永远无法比肩的 Python 权威，他为 CSC241/242 系列课程的内容制定了第一份大纲，而这是本教材最初的种子。Iyad Kanj 首开课程 CSC241，并无私地允许我使用他开发的材料。Amber Settle 是除我之外第一次使用本教程授课的教师，谢天谢地，她取得了巨大的成功，这个成功归功于她本身就是一个优秀的教师。在我所认识的人中，Craig Miller 最深入地思考并阐述了计算机科学的基本概念。通过和他之间的许多有趣的讨论，我获得了一些见解，本教程因此也受益匪浅。最后，Marcus Schaefer 对本教程一半以上的内容进行了彻底的技术审查，大大改进和完善了本教程的内容。

如果没有 Wiley 出版社教科书代理 Nicole Dingley 的建议，我的课程讲义将停留在讲义层面而不会编辑成书。Nicole 把我与 Wiley 出版社的编辑 Beth Golub 联系在一起。感谢 Beth 做出了一个勇敢的决定，选择信任一个拥有奇怪的名字并且没有教材写作经验的外国人来编写教科书。Wiley 出版社的高级设计师 Madelyn Lesure，以及我的朋友兼邻居 Mike Riordan，帮助我实现了简单整洁的正文设计。最后，Wiley 出版社的高级编辑助理 Samantha Mandel 不知疲倦地让我的各章草稿进入审阅和出版环节。在整个教材出版过程中，Samantha 一直是一个职业化和优雅的典范，她为这本教材提出了无数精彩的建议和意见，使得本教材更加出色。

这本书的最终版本只是表面上看起来与最初的草稿类似。相对于初始版本，教材最终版取得了长足的改善，这归功于数十位评审者（其中很多是匿名）。陌生人的善意使本教材变得更完美，而这也使得我对教材审阅过程有了新的认识。审阅者们不仅能发现问题，而且有提供解决方案的热情。我万分感谢他们认真而系统的反馈。一些审阅者（包括 David Mutchler（罗斯霍曼理工学院），提供了他的姓名和电子邮件给我以保持进一步通信联系）超越其职责范围，帮助挖掘深埋在我的早期草稿中的潜在问题。Jonathan Lundell 还对本教材最后一章提供了技术审阅。由于时间上的限制，我没能把收到的所有有价值的建议都纳入教材中，对教材中任何疏漏的责任完全由我自己承担。

我要特别感谢使用本教程第 1 版授课并给予我宝贵反馈意见的教师们：Ankur Agrawal（曼哈顿学院），Albert Chan（费耶特维尔州立大学），Gabriel Ferrer（汉德里克斯学院），David G. Kay（加利福尼亚大学欧文分校），Gerard Ryan（新泽西科技学院），Sridhar Seshadri（得克萨斯大学阿灵顿分校），Richard Weiss（常青州立大学），Michal Young（俄勒冈大学）等。我已经尽力在第 2 版中采纳他们的建议。

最后，我要感谢我的爱人 Lisa 和女儿 Marlena、Eleanor，感谢她们给予我的耐心。编写一本教材需要花费大量的时间，而这些时间只能来自家庭时间或睡眠时间，因为其他职业责任均有其设定的时间。编写这本教材花费的时间使得我常常无法参加家庭活动，或者由于睡眠不足而导致脾气不好。幸运的是，我有先见之明，在开始这个项目的时候领养了一只狗。虽然在家庭活动中的缺席带来了许多遗憾和失望，但这只名叫 Muffin 的狗无疑为我的家庭带来了更多的快乐……所以，还要感谢 Muffin。

关于作者

Ljubomir Perkovic 是芝加哥 DePaul 大学计算学院的副教授。他于 1990 年在纽约城市大学亨特学院获得了数学和计算机科学的学士学位，于 1998 年在卡内基-梅隆大学计算机科

学学院获得了算法、组合数学和优化的博士学位。

Perkovic 于 21 世纪初在 DePaul 大学开始教授程序设计入门系列课程。他的目的是与初级程序员分享开发人员开发一个很酷的新应用程序时的兴奋和喜悦。他把课程概念和现代应用程序开发中所使用的技术有机地融合在一起。他为这门课程所开发的材料构成了这本教程的基础。

他的研究方向包括计算几何、分布式计算、图论和算法以及计算思维。他对计算几何的研究让他获得了富布莱特研究学者奖和美国国家科学基金会项目（该项目研究如何在通识教育课程中扩大计算思维的应用）资助。

出版者的话

译者序

前言

第 1 章 计算机科学导论 1

1.1 计算机科学 1

1.1.1 计算机专业人员的工作 1

1.1.2 模型、算法和程序 2

1.1.3 必备的工具 3

1.1.4 什么是计算机科学 3

1.2 计算机系统 3

1.2.1 计算机硬件 3

1.2.2 操作系统 4

1.2.3 网络和网络协议 5

1.2.4 程序开发语言 5

1.2.5 软件库 6

1.3 Python 程序设计语言 6

1.3.1 Python 简史 6

1.3.2 构建 Python 开发环境 7

1.4 计算思维 7

1.4.1 一个示例问题 8

1.4.2 抽象和建模 8

1.4.3 算法 9

1.4.4 数据类型 9

1.4.5 赋值语句和执行控制结构 10

1.4.6 本章小结 11

第 2 章 Python 数据类型 12

2.1 表达式、变量和赋值语句 12

2.1.1 代数表达式和函数 12

2.1.2 布尔表达式和运算符 14

2.1.3 变量和赋值语句 16

2.1.4 变量名称 18

2.2 字符串 18

2.2.1 字符串运算符 19

2.2.2 索引运算符 21

2.3 列表和元组 22

2.3.1 列表运算符 22

2.3.2 列表是可变类型，字符串
是不可变类型 24

2.3.3 元组 24

2.3.4 元组和列表的方法 26

2.4 对象和类 28

2.4.1 对象类型 28

2.4.2 数值类型的有效值 29

2.4.3 数值类型的运算符 30

2.4.4 创建对象 32

2.4.5 隐式类型转换 32

2.4.6 显式类型转换 33

2.4.7 类方法和面向对象的
程序设计 34

2.5 Python 标准库 35

2.5.1 math 模块 35

2.5.2 fractions 模块 36

2.6 电子教程案例研究：海龟图形 37

2.7 本章小结 37

2.8 练习题答案 37

2.9 习题 39

第 3 章 命令式编程 43

3.1 Python 程序 43

3.1.1 我们的第一个 Python 程序 43

3.1.2 Python 模块 45

3.1.3 内置函数 print() 45

3.1.4 使用 input() 函数实现
交互式输入 46

3.1.5 eval() 函数 47

3.2 执行控制结构 47

3.2.1 单分支结构 48

3.2.2 双分支结构 51

3.2.3 循环结构	52	4.3.3 读取文本文件的模式	96
3.2.4 嵌套的控制流结构	55	4.3.4 写入文本文件	98
3.2.5 range() 函数	55	4.4 错误和异常	99
3.3 用户自定义函数	57	4.4.1 语法错误	100
3.3.1 我们自定义的第一个函数	57	4.4.2 内置异常	100
3.3.2 函数输入参数	58	4.5 电子教程案例研究: 图像文件	102
3.3.3 print() 与 return 的 比较	59	4.6 本章小结	102
3.3.4 函数定义实际上是 “赋值”语句	60	4.7 练习题答案	103
3.3.5 注释	61	4.8 习题	104
3.3.6 文档字符串	61	4.9 思考题	106
3.4 Python 变量和赋值语句	63	第 5 章 执行控制结构	108
3.4.1 可变类型和不可变类型	63	5.1 选择控制和 if 语句	108
3.4.2 赋值语句和可变性	65	5.1.1 三路以及多路分支	108
3.4.3 交换	66	5.1.2 条件的排列顺序	110
3.5 参数传递	67	5.2 for 循环和迭代模式	111
3.5.1 不可变类型参数传递	67	5.2.1 循环模式: 迭代循环	111
3.5.2 可变类型参数传递	69	5.2.2 循环模式: 计数器循环	112
3.6 电子教程案例研究: 自动化 海龟图形	70	5.2.3 循环模式: 累加器循环	114
3.7 本章小结	70	5.2.4 其他类型的累加	115
3.8 练习题答案	70	5.2.5 循环模式: 嵌套循环	117
3.9 习题	73	5.3 深入研究列表: 二维列表	119
3.10 思考题	74	5.3.1 二维列表	119
第 4 章 文本数据、文件和异常	77	5.3.2 二维列表和嵌套循环模式	120
4.1 深入研究字符串	77	5.4 while 循环	122
4.1.1 字符串表示	77	5.5 更多循环模式	124
4.1.2 深入研究索引运算符	79	5.5.1 循环模式: 序列循环	124
4.1.3 字符串方法	80	5.5.2 循环模式: 无限循环	125
4.2 格式化输出	83	5.5.3 循环模式: 循环和折半	126
4.2.1 函数 print()	83	5.6 其他迭代控制语句	127
4.2.2 字符串方法 format()	84	5.6.1 break 语句	127
4.2.3 按列对齐排列数据	86	5.6.2 continue 语句	128
4.2.4 获取与格式化日期和时间	89	5.6.3 pass 语句	129
4.3 文件	91	5.7 电子教程案例研究: 图像处理	129
4.3.1 文件系统	91	5.8 本章小结	130
4.3.2 打开和关闭文件	92	5.9 练习题答案	130
		5.10 习题	133
		5.11 思考题	134

第 6 章 容器和随机性	140	7.1.5 与函数调用相关的名称空间	173
6.1 字典	140	7.1.6 名称空间与程序栈	174
6.1.1 用户自定义索引作为字典的 动机	140	7.2 全局名称空间和局部名称空间	177
6.1.2 字典类属性	141	7.2.1 全局变量	178
6.1.3 字典运算符	143	7.2.2 局部作用范围的变量	178
6.1.4 字典方法	144	7.2.3 全局作用范围的变量	179
6.1.5 字典作为多路分支 if 语句的 替代方法	146	7.2.4 在函数中改变全局变量	180
6.1.6 字典作为计数器集合	147	7.3 异常控制流	181
6.1.7 元组对象可以作为字典 的键	149	7.3.1 异常和异常控制流	182
6.2 集合	150	7.3.2 捕获和处理异常	183
6.2.1 使用 set 构造函数移除 重复项	151	7.3.3 默认异常处理程序	184
6.2.2 set 运算符	152	7.3.4 捕获给定类型的异常	184
6.2.3 set 方法	153	7.3.5 多重异常处理程序	185
6.3 字符编码和字符串	154	7.3.6 控制异常流	186
6.3.1 字符编码	154	7.4 模块作为名称空间	189
6.3.2 ASCII	154	7.4.1 模块属性	189
6.3.3 Unicode	156	7.4.2 导入模块时发生了什么	190
6.3.4 Unicode 字符的 UTF-8 编码	158	7.4.3 模块搜索路径	190
6.4 random 模块	158	7.4.4 顶层模块	192
6.4.1 选择一个随机整数	159	7.4.5 导入模块属性的不同方法	193
6.4.2 选择一个随机“实数”	160	7.5 类作为名称空间	195
6.4.3 随机混排、挑选和抽样	161	7.5.1 一个类是一个名称空间	195
6.5 电子教程案例研究：机会游戏	161	7.5.2 类方法是在类名称空间中 定义的函数	196
6.6 本章小结	162	7.6 电子教程案例研究：使用调试器 进行调试	197
6.7 练习题答案	162	7.7 本章小结	197
6.8 习题	165	7.8 练习题答案	197
6.9 思考题	166	7.9 习题	199
第 7 章 名称空间	171	7.10 思考题	200
7.1 函数封装	171	第 8 章 面向对象的程序设计	202
7.1.1 代码重用	172	8.1 定义新的 Python 类	202
7.1.2 模块化	172	8.1.1 类 Point 的方法	202
7.1.3 封装	172	8.1.2 类和名称空间	203
7.1.4 局部变量	172	8.1.3 每个对象都有一个关联的 名称空间	204
		8.1.4 类 Point 的实现	204
		8.1.5 实例变量	206
		8.1.6 实例继承类属性	206

8.1.7	类定义的一般格式	207	9.1.1	组件 Tk: GUI 窗口	246
8.1.8	编写类的文档	208	9.1.2	组件 Label: 显示文本	247
8.1.9	类 Animal	208	9.1.3	显示图像	248
8.2	用户自定义类示例	210	9.1.4	布局组件	249
8.2.1	构造函数重载	210	9.1.5	将组件布局为表格	251
8.2.2	默认构造函数	211	9.2	基于事件的 tkinter 组件	253
8.2.3	扑克牌类	212	9.2.1	Button 组件及事件处理程序	253
8.3	设计新的容器类	213	9.2.2	事件、事件处理程序和 mainloop()	255
8.3.1	设计一个表示一副扑克牌的类	213	9.2.3	Entry 组件	256
8.3.2	实现 Deck 类	214	9.2.4	Text 组件和绑定事件	258
8.3.3	容器类 Queue	215	9.2.5	事件模式和 tkinter 类 Event	259
8.3.4	实现 Queue 类	216	9.3	设计图形用户界面	261
8.4	运算符重载	217	9.3.1	组件 Canvas	261
8.4.1	运算符是类方法	218	9.3.2	作为组织容器的组件 Frame	264
8.4.2	使 Point 类对用户友好	220	9.4	面向对象的图形用户界面	265
8.4.3	构造函数和 repr() 运算符之间的约定	221	9.4.1	GUI 面向对象程序设计基本知识	266
8.4.4	使 Queue 类对用户友好	223	9.4.2	把共享组件赋值给实例变量	268
8.5	继承	225	9.4.3	把共享数据赋值给实例变量	269
8.5.1	继承类的属性	225	9.5	电子教程案例研究: 开发一个计算器	271
8.5.2	类定义的一般格式	227	9.6	本章小结	271
8.5.3	重写父类方法	227	9.7	练习题答案	271
8.5.4	扩展父类方法	229	9.8	习题	275
8.5.5	通过继承 list 实现一个 Queue 类	231	9.9	思考题	275
8.6	用户自定义异常	232	第 10 章	递归	278
8.6.1	抛出一个异常	232	10.1	递归简介	278
8.6.2	用户自定义异常类	233	10.1.1	调用自身的函数	278
8.6.3	改进类 Queue 的封装	234	10.1.2	停止条件	279
8.7	电子教程案例研究: 索引和迭代器	234	10.1.3	递归函数的特性	280
8.8	本章小结	235	10.1.4	递归思想	281
8.9	练习题答案	235	10.1.5	递归函数调用和程序栈	282
8.10	习题	237	10.2	递归示例	284
8.11	思考题	239			
第 9 章	图形用户界面	246			
9.1	tkinter 图形用户界面开发基本知识	246			