Mc Graw Hill Education 大学计算机教育国外著名教材系列 影印版

# Programming Languages
## Principles and Paradigms
## Second Edition

# 编程语言
## 原理与范型（第2版）

Allen B. Tucker
Robert E. Noonan 著

Mc Graw Hill

清华大学出版社

大学计算机教育国外著名教材系列（影印版）

# Programming Languages

## Principles and Paradigms

Second Edition

# 编程语言

## 原理与范型

（第2版）

Allen B. Tucker

Robert E. Noonan

清华大学出版社

北京

Allen B. Tucker, Robert E. Noonan
**Programming Languages: Principles and Paradigms, Second Edition**
EISBN: 0-07-286609-8



**图书在版编目(CIP)数据**

编程语言：原理与范型 = Programming Languages: Principles and Paradigms: 第 2 版: 英文 / （美） 塔克（Tucker, A. B.），（美）努南（Noonan, R. E.）著. —影印本. —北京：清华大学出版社，2009.5
（大学计算机教育国外著名教材系列（影印版））
ISBN 978-7-302-19806-2

Ⅰ. 编… Ⅱ. ①塔…②努… Ⅲ. 程序语言－高等学校－教材－英文 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2009）第 045642 号

**责任印制：** 李红英

**出版发行：** 清华大学出版社　　**地　　址：** 北京清华大学学研大厦 A 座
http:/ /www.tup.com.cn　　**邮　　编：** 100084
**社　总　机：** 010-62770175　　**邮　购：** 010-62786544
**投稿与读者服务：** 010-62776969, c-service@tup.tsinghua.edu.cn
**质　量　反　馈：** 010-62772015, zhiliang@tup.tsinghua.edu.cn
**印 刷 者：** 北京密云胶印厂
**装 订 者：** 北京市密云县京文制本装订厂
**发 行 者：** 全国新华书店
**开　　本：** 185×230　　**印张：** 38.25
**版　　次：** 2009 年 5 月第 1 版　　**印　　次：** 2009 年 5 月第 1 次印刷
**印　　数：** 1~3000
**定　　价：** 59.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：010-62770177 转 3103　　产品编号：025395-01

# 出 版 说 明

进入 21 世纪，世界各国的经济、科技以及综合国力的竞争将更加激烈。竞争的中心无疑是对人才的竞争。谁拥有大量高素质的人才，谁就能在竞争中取得优势。高等教育，作为培养高素质人才的事业，必然受到高度重视。目前我国高等教育的教材更新较慢，为了加快教材的更新频率，教育部正在大力促进我国高校采用国外原版教材。

清华大学出版社从 1996 年开始，与国外著名出版公司合作，影印出版了“大学计算机教育丛书（影印版）”等一系列引进图书，受到国内读者的欢迎和支持。跨入 21 世纪，我们本着为我国高等教育教材建设服务的初衷，在已有的基础上，进一步扩大选题内容，改变图书开本尺寸，一如既往地请有关专家挑选适用于我国高校本科及研究生计算机教育的国外经典教材或著名教材，组成本套“大学计算机教育国外著名教材系列（影印版）”，以飨读者。深切期盼读者及时将使用本系列教材的效果和意见反馈给我们。更希望国内专家、教授积极向我们推荐国外计算机教育的优秀教材，以利我们把“大学计算机教育国外著名教材系列（影印版）”做得更好，更适合高校师生的需要。

清华大学出版社

# Preface
# 前　言

自本书第1版于1999年出版以来，编程语言的研究已得到迅猛发展。例如，从CS1开始，Java已经成为计算机科学课程的一门重要语言。敏捷编程与软件设计如影随形，且其语言习惯有别于传统的程序。用正规的方法进行软件设计已渐入主流，且作用显著。

有鉴于此，新版希望能够适应在当前和未来编程语言设计过程中所伴随的激励和新挑战。例如，新版对全部4种程序设计范例及其所用的语言在广度和深度上都有进一步的提高，如表0-1所示。

表0-1　语言覆盖范围

| 范例 | 第1版 | 第2版 |
| --- | --- | --- |
| 命令式(第12章) | | C、Ada、Perl |
| 面向对象(第13章) | Java | Java、Smalltalk、Pythob |
| 函数式(第14章) | Scheme、Haskell | Scheme、Haskell |
| 逻辑式(第15章) | Prolog | Prolog |

新版第二个主要的变化是大大丰富了第2~11章关于语言设计原理的内容。我们使用不太正式的描述风格，增加了新的来自流行编程语言(如Python 和 Perl)的例子。此外，对于已经不再广泛应用的语言(如Pascal和Modula)，新版多已略去。

第2、4、5、7、9章主要讲述编程语言的核心原理——语法、名称、类型、语义和函数。这几章与第1版相比较，为上机学习这些原理提供了更广泛、更深入的语言和范例。

如果您希望了解语法、类型系统、语义、函数和存储管理的具体实现原理，可以在第3、6、8、10和11章中找到这些材料。读者可以有选择地学习这几章，以丰富相关的核心原理。例如，通过学习第3章中编译器的词汇和语法段，可加深和巩固对第2章的学习。注意，可以跳过部分或全部的章节，特别是初学编程

语言的学生。

有3章包括了需要较多数学背景知识的可选部分。附录B提供了这些章节中主要的离散数学问题和概念，供需要快速复习的学生使用。

最后，第16、17和18章详细介绍了事件处理、并发性和程序正确性。第1版涵盖其中的两部分(事件处理和并发性)，而新版加入了关于程序正确性的介绍。

总之，新版在编程语言的原理、范例和专题方面涵盖的范围更广、更深。本书共有18章。由于不同的教师对编程语言课程强调的重点有不同的看法，所以本书提供了多种选择。

## 0.1 提示

本书强调在编程语言设计中对于关键问题的全面实践操作。它为老师和学生提供了理论与实现相结合的经验。基于实现的经验包括C语言简单子集的设计和实施的上机操作，这被称作Clite。为便于参考，在附录A中给出了完整定义。

如上所述，本版对于主要编程范例进行了扩展。我们相信，为了掌握一个范例，学生们必须积极运用范例去解决编程问题。例如，如果学生没有函数式编程的经验，建议他们充分学习Scheme或者Haskell，完成一个说得过去的编程项目。引用一个评论者的评语：

*为了明白一个范例，您自己必须实现这个范例。*

另一方面，如果您的预备课程离散数学或者AI课程已经包括函数式程序设计，可以选择跳过这一章，重点学习其他的范例或专题。

## 0.2 本书的结构

图0-1解释了正文如何分为3个主要的部分：

- 原理。
- 范例。
- 专题。

在第1部分，第2、4、5、7、9章包括5个核心原理—— 语法、名称、类型、语义和函数。该部分剩余章节(第3、6、8、10、11章)有助于加深对这些主题的理解。

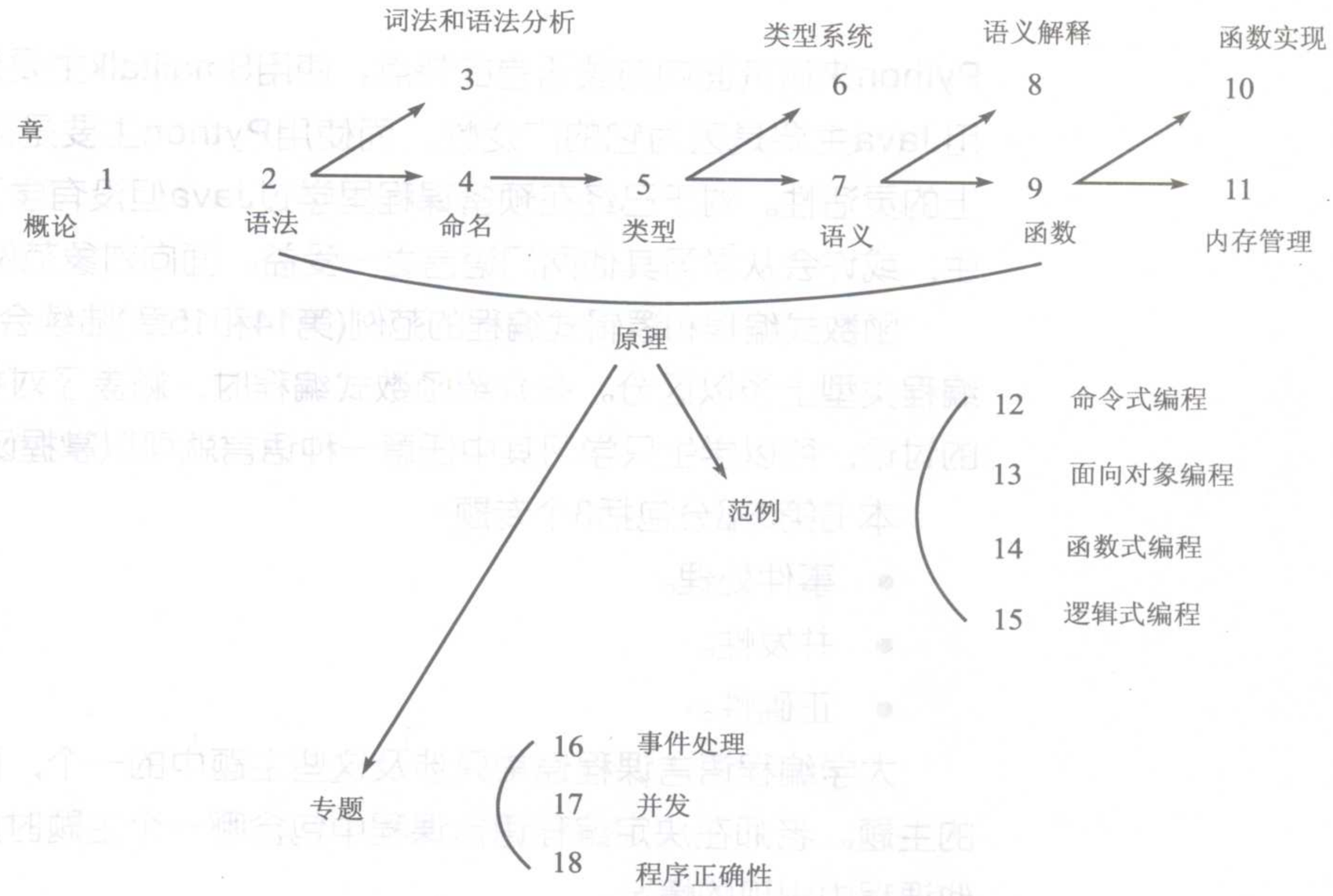


图0-1 各章之间的内容关系

另一方面，高年级或研究生课程可能会包括其中的一些内容，而没有强调的材料在后续的章节中介绍。

本书的第2部分包含4个主要的程序范例：

- 命令式编程。
- 面向对象编程。
- 函数式编程。
- 逻辑式编程。

这些章之间相对独立并且可以按任意顺序学习。尽管如此，对于一个学期的课程安排，除了原理之外，一般只包括命令式编程和面向对象。当然，范例的选择可能会根据教师的侧重和该学期课程内容的不同而不同。例如，如果该学期已经有面向对象设计课程，就可以跳过第13章。

新版中的第12章是全新的内容，这一章阐述了在C、Ada和Perl 等3种不同的语言中命令式编程中的关键特点，选择C语言是因为它说明了弱类型引起的问题。相反，Ada是强类型，提供了和C语言生动的对比。最后，选择Perl是由于它阐述了一种动态类型脚本语言。对讲授第12章的教师，建议在安排趣味编程项目时，包含至少一门有足够深度的语言，并且学生不熟悉这门语言。

第13章在第1版的基础上进行了较多改写。新版中，使用Java、Smalltalk和

Python来研究面向对象语言的特点。使用Smalltalk主要是因为它的简单性，使用Java主要是因为它的广泛性，而使用Python主要是因为它在脚本应用程序上的灵活性。对于已经在预备课程里学过Java但没有学习过面向对象设计的学生，或许会从学习其他两门语言之一受益。面向对象范例今后一直会很重要。

函数式编程和逻辑式编程的范例(第14和15章)陆续会在其应用程序和各自的编程类型上予以区分。在介绍函数式编程时，涵盖了对并行Scheme和Haskell的讨论，所以学生只学习其中任意一种语言就可以掌握函数式编程的思想。

本书第3部分包括3个专题：

- 事件处理。
- 并发性。
- 正确性。

大学编程语言课程通常只涉及这些主题中的一个，而没有时间去涉及其他的主题。老师在决定编程语言课程中包含哪一个主题时应考虑到这些主题在其他课程中出现的情况。

第16和17章的主题是事件处理和并发性，叙述了反常控制问题，在编程语言的学习中需要认真对待这个问题。在编程应用中，特别是在科学计算和嵌入系统方面，二者的作用日渐凸现。学生们需要学习的主要内容包括：通信、死锁、信息传递、非确定性、事件处理、交互处理通信，以及种类繁多的具体应用(如操作系统、GUI交互和家用报警系统)中的某些个案。

最后，第18章关于正确性的内容将让人耳目一新，这一章主要分析了编程语言在支持常用方法上的发展。例如， Spark Ada的开发人员宣称使用“由架构保证正确性”的技术使编程错误数目降低了99%，并使生产率提高了3~5倍。我们认为未来对编程语言而言，软件设计中对形式方法的支持将日渐重要(很像现今面向对象编程重要一样)，因此，希望老师能在编程语言课程中加入这一章。

第18章一开始回顾了公理语义学及其在命令式编程检验中的应用。这一章涵盖了在面向对象语言中“契约设计(design by contract)”的概念及其使用Java建模语言的具体应用，并延伸了这一理论。本章也涉及了结构化归纳的概念及其在检验函数式程序的正确性方面的应用。

需要说明的是，后面章节的一些问题应该在其他课程中学习，而不是仅在编程语言和范例中学习。比如，并发性、事件驱动编程和程序正确性都应该作为一门独立的课程进行学习。此外，在第3、6、8、10和11章中很多的内容可以作为编译课程的基础。因为很多大学课程安排中没有提供这些主题的专门课程，本书只是在编程语言课程中作入门性的简单介绍。

## 0.3 本书预备知识和学习指导

在学习这门课之前应该至少已经完成一门预科课程或者学习过数据结构。在数据结构课程中，应该已经熟悉了链表、堆栈、稀疏矩阵和哈希表。

此外，在第3、6、8、16和17章任何含有程序执行的内容中，Java知识是学习的先决条件。如果没有学习过Java，也应该通过数据结构课程来熟悉C、C++或C#方面的知识。

建议学生准备一个好的Java指南、参考手册和编程环境。这些在网络上很容易获得(如http://java.sun.com/docs/books/tutorial/)。我们讨论的某些内容需要Java 1.4和Java 1.5引入的新特性的支持。这些特性在文中已清楚标明。

我们也希望学生有一定的数学知识，即初等离散数学和离散结构课程的相关知识。尽管这不是必需的，但是对于要学习数学因素浓厚的某些章节(第3、6、8、10和18章)的学生而言还是非常必要的。附录B中回顾了基本的函数、集合、逻辑和证明的内容。

本书与Computing Curricula 2001 [CC 2001]的介绍是一致的。本书涵盖了Liberal Arts Model Curriculum [Walker and Schneider, 1996]及其2005年未定稿版本中描述的编程语言课程中所涉及的所有问题。

本书涵盖了Computing Curricula 2001的核心知识部分关于编程语言的所有主题(从1到11部分)，还涉及该部分的其他主题，例如，事件驱动和并发编程(PF6)、内存管理(OS5)、函数及逻辑式编程(IS)和软件工程(SE)。相对于Computing Curricula 2001，本书对每个主题都用了更多的篇幅进行讲解。

## 0.4 编程语言资源和Web站点

本文所涉及的软件都可在Java 1.5或者更高的版本中执行，而且我们已经使用SUN公司开发的Java 1.5执行过书中的程序。表0-2是我们推荐的一系列网络资源，包括书中涉及的主要编程语言的学习指南和其他信息。

表 0-2　编程语言与 Web 站点

| 语　言 | Web资源地址 |
|---|---|
| Ada | www.gnu.org |
| C、C++ | www.gnu.org |
| Haskell | www.haskell.org |
| Java 1.5 | www.java.sun.com |
| Perl | www.perl.com |

(续表)

| 语 言 | Web 资源地址 |
| --- | --- |
| Prolog | www.python.org |
| Python | www.drscheme.org |
| Scheme | www.drscheme.org |
| Smalltalk | www.squeak.org |

与该书配套的软件和相关的资料可以在www.mhhe.com/tucker站点找到，具体包括包括：

- Clite的Java语法、类型系统和语义的完整Java实现过程。
- 本文所有可运行程序的下载包。
- 与本书配套的授课幻灯片。
- 课后作业答案(教师可通过密码认证获得)。

## 0.5 致谢

在写作过程中，很多人给予了指导。James Lu是第1版初稿的最主要合作者。William学院的Bill Bynum、Holy Cross学院的Mary和Laurie King分别撰写了第4章和第8章。William学院的David Coppit介绍了第18章中出现的校验树的使用。Bowdoin学院和William and Mary学院的学生为第1版和第2版的早期版本作出了贡献。还要特别提到的是，Doug Vail对一些富于挑战性的问题提出了解决的办法。Wyatt Dumas帮助重写了第2版中的程序，并在其中两章的内容上使我们受益匪浅。我们要感谢所有的审阅者：

| | |
| --- | --- |
| Qi Cheng | *Oklahoma* 大学 |
| Rainey Little | *Mississippi* 州立大学 |
| Jay-Evan J. Tevis | *Auburn* 大学 |
| John Hannan | *Pennsylvania* 州立大学 |
| Neelam Soundarajan | *Ohio* 州立大学 |
| Robert van Engelen | *Florida* 州立大学 |
| Shannon Tauro | *California* 大学 *Irvine* 分校 |
| Gloria Melara | *California* 州立大学 *Northridge* 学院 |
| Amer Diwan | *Colorado* 大学 *Boulder* 学院 |
| Susan Gauch | *Kansas* 大学 |
| Henri Casanova | *California* 大学 *San Diego* 学院 |
| Cristina V. Lopes | *California* 大学 *Irvine* 学院 |

Salih Yurttas　*Texas A&M* 大学
Roman W. Swiniarksi　*San Diego* 州立大学
Amar Raheja　*California State Polytechnic* 大学 *Pomona* 学院
Franck Xia　*Missouri* 大学 *Rolla* 学院
Rajendra K. Raj　*Rochester* 技术学院
Randall D. Beer　*Case Western Reserve* 大学
Robert M. Cubert　*Florida* 大学
Liang Cheng　*Lehigh* 大学
David Hemmendinger　*Union* 学院

正是他们仔细的阅读、富有建设性的评审和集体智慧，极大地推动了本书第1版和第2版的出版。也正是通过他们集体而卓越的洞察力使得本书第1和第2版有了本质的提高。作者还要特别感谢Union学院的David Hemmendinger为本书所作的认真编辑和详尽建议，其中大部分已经融入到书中。

最后，我还要感谢我们的编辑Rebecca Olson和Alan Apt，感谢他们的见识、指导和支持。他们非凡的技巧使得第2版有了更大的发展。

Allen B. Tucker　*Bowdoin* 学院
Robert E. Noonan　*William and Mary* 学院

# 作者介绍

**Allen B. Tucker** 是 Bowdoin 学院计算机科学系的教授，曾获得 Wesleyan 大学的数学学士学位，Northwestern 大学的计算机科学硕士学位。

Tucker 教授已出版了有关编程语言、软件设计、自然语言处理及课程设计等方面的著作。他是乌克兰 Ternopil Academy of National Economy 的 Fulbright 讲席教授，新西兰 Canterbury 大学的 Erskine 访问学者和法国 Esigelec 大学的访问学者，他还是 ACM 的会员。

**Robert E. Noonan** 是 William and Mary 学院的计算机科学系教授，已经从事计算机教学 30 多年。他在 Providence 学院获得数学学士学位，在 Purdue 大学获得计算机科学硕士和博士学位。

他已出版过有关编程语言、编译器构建和软件工程方面的著作，他是 ACM、SIGPLAN、SIGCSE 以及 LACS Consortium 的会员。

# Contents

# 目　录