Programming Languages
Concepts & Constructs (Second Edition)

程序设计语言
概念和结构
（原书第2版）

(美) Ravi Sethi 著
贝尔实验室
裘宗燕 等译

机械工业出版社
China Machine Press
Addison-Wesley

HZ BOOKS
华章教育

国外经典教材

Classical Texts From Top Universities

程序设计语言
概念和结构

[redacted]

[redacted]

HZ BOOKS
华章教育
北京华章图文信息有限公司

国外经典教材 Classical Texts From Top Universities

程序设计语言
概念和结构 (原书第2版)

Programming Languages:
Concept & Constructs *(Second Edition)*

(美) Ravi Sethi 贝尔实验室 著
裘宗燕 等译

机械工业出版社
China Machine Press

本书是国外比较成功的一本讨论程序设计语言的教科书，已在一些学校使用多年。书的主要内容包括：引论、命令式程序设计、面向对象的程序设计、函数式程序设计、其他程序设计范型以及语言的描述六大部分。本书适合作为计算机及其相关专业本科高年级学生的教材或教学参考书，或作为研究生的基础课程教材或参考书，也适合其他相关的技术人员参考。本书的学习基础是学过用过一种或几种程序设计语言（最好是Pascal/C/C++），有一定程序设计经验，并对数据结构等计算机基础知识有所理解。

本书版权登记号：图字：01-2001-2197

图书在版编目（CIP）数据

程序设计语言：概念和结构（原书第2版）/（美）塞西（Sethi, R.）著；裘宗燕等译. -北京：机械工业出版社，2002.2
（国外经典教材）
书名原文：Programming Languages: Concepts & Constructs, 2E
ISBN 7-111-09431-x

Ⅰ.程… Ⅱ.①塞… ②裘… Ⅲ.程序语言－教材 Ⅳ.TP312

中国版本图书馆CIP数据核字（2001）第084567号

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：焦胜才 陈贤舜
北京忠信诚印刷厂印刷·新华书店北京发行所发行
2002年2月第1版第1次印刷
787mm×1092mm 1/16·30.75印张
印数：0 001-5 000册
定价：45.00元

出版者的话

文艺复兴以降，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；也正是这样的传统，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，不仅擘划了研究的范畴，还揭橥了学术的源变，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、从业人员较少的现状下，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图文信息有限公司较早意识到“出版要为教育服务”。自1998年始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann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Tanenbaum，Stroustrup，Kernighan，Jim Gray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、研究及庋藏。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襄助，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，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，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，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步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步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：针对本科生的核心课程，剔抉外版菁华而成“国外经典教材”系列；对影印版的教材，则单独开辟出“经典原版书库”；定位在高级教程和专业参考的“计算机科学丛书”还将保持原来的风格，继续出版新的品种。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

“国外经典教材”是响应教育部提出的使用外版教材的号召，为国内高校的计算机本科教学

度身订造的。在广泛地征求并听取丛书的“专家指导委员会”的意见后，我们最终选定了这20多种篇幅内容适度、讲解鞭辟入里的教材，其中的大部分已经被M.I.T.、Stanford、U.C. Berkley、C.M.U.等世界名牌大学采用。丛书不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

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，但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。教材的出版只是我们的后续服务的起点。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，我们的联系方法如下：

电子邮件：hzedu@hzbook.com
联系电话：（010）68995265
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

专家指导委员会

（按姓氏笔画顺序）

译 者 序

IEEE/ACM的1991教程将“程序设计语言”课程列为计算机及其相关专业本科生必修课程。在IEEE/ACM联合拟定的2001教程草案中，也将“程序设计语言”列为教育计划的14个基本知识领域之一，并特别强调了它的重要性。本书就是国外比较成功的一本讨论程序设计语言的教科书，已在一些学校使用多年（这里是第2版）。本书作者是贝尔实验室计算科学研究中心主任，在程序设计语言、编译技术等方面颇有建树，在程序设计语言的教学方面也有丰富经验。本书取材比较合理，既讨论了许多实际技术问题，以帮助人们认识程序设计语言的方方面面；也适当地介绍了一些问题的理论基础。书的内容组织也比较合理。本书适合作为计算机及其相关专业本科高年级学生的教材或教学参考书，或作为研究生的基础课程教材或参考书，也适合其他热爱计算机科学技术、希望进一步提升自己的计算机基础和能力的人阅读参考。本书的学习基础是学过用过一种或几种程序语言（最好包括Pascal/C/C++），有一定程序设计经验，并对数据结构等计算机基础知识有所理解。

本书以介绍常规的命令式语言及其面向对象的发展为重点，讨论了常规语言各方面的基本问题，包括语言描述，控制结构、数据类型以及它们的实现机理，过程抽象及其实现，数据抽象和面向对象的有关特征及其实现，并行机制的各种问题。也用相当的篇幅介绍了程序设计的其他范型，主要是函数式程序设计、逻辑程序设计，以及支持这些范型的语言。学习了解这些知识，对于计算机及其相关专业学生的知识完整性也是非常重要的。从这些非常规的范型中可以看到许多很有价值的思想。实际上，无论是过去、现在、还是将来，其他程序设计范型的许多有价值的思想都已经/正在/将以各种形式被常规程序设计实践所接纳和使用。了解在更大范围里的程序设计实践情况、对于在计算机领域中的创新思维也是极端重要的，无论是从事实践性工作还是研究性工作。

本书还用了一定篇幅，简略讨论了与程序语言有关的一些理论问题，包括程序语言的语法、语义模型、类型，等等。虽然大部分计算机领域的工作者不会去从事理论研究工作，但也应该看到理论对于实际工作的重要作用。按我的个人观点，理论至少可以帮助人擦亮眼睛，使我们能看清楚那些隐藏在繁杂的实际细节背后的根本原因，使我们在从事实践性的工作时心里有底。对理论问题的重视不够，至少说，已经在一定程度上阻碍了我国计算机事业的发展，阻碍了许多很有潜力的个人的发展。虽然改变计算机教育的现状，推动我国计算机事业的发展不是一朝一夕之事，但是，每个立志投身于这一领域的个人还是能尽量为自己的未来前途多做一些事情。加强个人的理论素养就是最基本的问题之一。本书提供了有关方面的一些基本知识，可以作为相关学习的起点。

对于大部分计算机工作者、大部分软件人员、大部分计算机及其相关专业的学生而言，程序设计语言就是我们所面对的计算机。因此，理解计算机、掌握计算机技术的一个重要方面就是理解和掌握程序设计语言。程序语言规定了我们在描述要解决的问题时所能使用的手段，规定了这种描述的形式及其意义。在另一个层次上，一种特定的程序语言还对我们在解决问题时可能采用的技术和方式限定了一个大的边界。

示对象的应用。

第5章的讨论围绕着命令式程序设计的核心，也就是像Pascal或C一类语言的内部东西。有关的论题包括：过程的源代码正文与过程活动的对比，参数传递机制，作用域规则和存储管理等。

在本书中，只要可能，就采用Pascal去展示命令式程序设计的有关问题。Pascal对于第3～5章而言已经足够了。另一种可能的选择是用C语言。

第三部分：面向对象程序设计

随着程序变得越来越大，程序设计的自然单位也变成了一组组的数据和操作。程序结构概念的发展产生了一系列的术语，包括模块、用户定义类型（例如堆栈）以及类（在面向对象的程序设计中）。

第6章开始讨论基于过程、模块和类的程序设计。这些概念服务于不同的需要，也能够互相组合一起使用：在实现模块或者类的操作时总需要使用过程；模块能够被用作带类的程序的源代码中的静态部分。Pascal的某些版本支持模块概念，因此它们就可以用于第6章的前半部分。C++是C的一种扩充，第6章里将会介绍它。

第7章中的计算模型是独立的对象，这些对象通过互相传递消息的方式交互。这一章的前三分之一是有关面向对象程序设计的一般性讨论，用了一个在C++和Smalltalk里可以有相似实现的实例。其余部分是有关C++和Smalltalk的独立材料，以便能用这两种语言中的任何一种来研究面向对象程序设计的问题。基于教师的反馈，这里将C++放在Smalltalk之前，与原版本相比，交换了位置。面向对象的程序设计用C++和Smalltalk阐释，是因为它们代表了两条不同的途径。

第3～7章里的所有概念都可以用C++语言阐述。因此，也可以直接向学生介绍C++，无须先经过C语言。

第四部分：函数式程序设计

函数式程序设计是一种非常值得学习的程序设计风格，无论是就其自身，是作为一种研究类型等概念的媒介，还是作为一种描述语言的技术。第8章强调的重点还是概念，第9章和第10章强调程序设计风格，第13章则是语言描述。其中的计算模型是基于一个表达式解释器，表达式通过将函数应用到一些子表达式上而构成。

第8章强调的是概念。函数式语言的简单性使它们可以非常方便地用于介绍诸如值、类型、名字和函数一类的概念。这种简单性结果源自对表达式和值的强调，与作为其基础的机器无关。本章讨论函数式语言的公共基础问题，以ML作为工作语言。

ML和Lisp之间的根本性差别在于ML是有类型的，类型的影响遍及整个语言。第9章用ML阐述函数和数据类型的使用。在函数式程序设计中，函数作为一级成员（一等公民），处在与其他值同等的地位。这种一级地位使我们可以构造出各种威力强大的针对数据集合的操作。

函数式程序设计始于Lisp。Lisp的程序和数据都用表来表示，甚至这种语言的名字本身也就是“表处理系统（List Processor）”的缩写形式。这种对于表的一致使用方式使得Lisp语言以其可扩充性而闻名于世。第10章将探索表的使用，以Lisp的方言Scheme作为基础。

另请参看第13章，那里包含了对Scheme的一个小子集的解释器。还有第14章，其中讨论了lambda演算。

第五部分：其他范型

在第11章中，逻辑式程序设计是结合Prolog一起讨论的。逻辑式程序设计处理的是关系而不是函数。对于合适的问题，采用这种范型写出的程序短小精悍，由规则和事实组成。这种语言利用给定的规则和事实，推导出对于查询的回答。

在第12章用Ada语言展示了并发程序设计的情况。也可以采用另一种方式，在面向对象的程序设计之后讨论并发程序设计，通过给每个对象一个计算线程，就能形成许许多多的进程。目前的组织方式是将函数式程序设计放在并发程序设计之前。

第六部分：语言的描述

第13章有关语言描述的方法介绍是针对专业人员写的。这些方法范围广泛，从为语言翻译而用的属性，到用于类型推理的逻辑规则，再到用于澄清微妙的语言问题的解释器。

要描述一种语言，可以为它专门写一个定义性解释器，这种解释器的目的就是定义被解释的语言，效率并不是这里关心的问题。McCarthy用Lisp语言写出了Lisp语言的最早的定义性解释器，在语言描述方面仍有重要意义，因此，这里的语言描述问题使用Lisp的方言Scheme。第13章为Scheme的一个小子集开发了一个解释器。

lambda演算是函数式程序语言在智慧方面的先驱者。lambda演算具有极其简单的语法形式，这使它被广泛用做研究语言的媒介。第14章介绍了lambda演算的一些变体，从纯的无类型lambda演算一直讨论到有类型的lambda演算。

第15章是对本书中所用的各种语言的一个总结。

RS

目前，世界上流行的程序语言有许多种，这种情况将来也不会改变，因为这是由我们所面临的需要解决的计算问题的五彩缤纷所决定的。任何语言都不可能是最完美的工具，不可能在处理任何问题时都是最锐利的宝剑。每种语言都有其最强的地方，也有其薄弱之处。当我们面对一项具体工作时，即使有各种环境因素的制约，也还是应该尽可能地去选择最适合这个问题的语言。一个计算机工作者需要理解掌握多种程序设计语言，只有这样，在遇到具体问题时才可能做出更合理的选择。为此，我们就需要学习多种程序设计语言，需要在实际工作中，针对具体问题，对各种可能用的语言进行评价和比较。而要能正确地做这种评价比较，就需要有对于语言的更深入理解。

各种常规语言有许多共同点，也有许多差异。许多共同点和差异在程序的表面形式上表现得并不明显。有些程序机制的引入将引起语言实现模型的根本性变化，例如支持递归，支持数组的越界检查，支持动态的类型检查，支持各种面向对象的概念，等等。这些都会对我们写程序的方式，写出的程序的效率和灵活性产生重要影响。为什么会出现这种情况？其根源就在于程序语言与常规计算机的深层关系，在于将这些语言特征映射到常规计算机的方式和技术。理解了这些问题，不但能够帮助我们做出更合理的语言选择，也使我们能更有把握地选择所使用的程序机制与结构，不管在哪种具体语言里写程序，这些选择也是极端重要的。对于本书有关知识的学习将能在这个方面给我们以帮助。

在实际中，新的语言层出不穷，学习新语言会成为许多计算机工作者毕生都要做的事情。由于新的应用问题（太多了，数不胜数），新的程序运行环境（例如网络环境，并行环境等），对于程序设计实践的新认识（例如，面向对象），理论研究的新成果，甚至是为了个别公司自身的商业利益，如此等等，这些都可能导致新语言的出现。学习程序设计语言课程或相关书籍有一个重要作用，那就是能使人更容易看清自己正在学习的语言的关键性特征，也更容易看清楚语言中的陷阱（没有语言能完全避免这些东西），使语言学习更有成效，也使人能将新学习的语言更快更好地应用于实际。

换一个角度，从更不功利的观点出发去看问题。程序设计语言是计算机科学技术领域中的一个重要分支，在过去的50年里，人们在这个领域中的研究和开发取得了极其丰硕的成果。这些成果，作为计算机科学技术研究的智力产出，也是非常值得了解和学习的。即使作为一种欣赏，观看、领略那些从事程序设计语言研究和实践的先行者的各种智力的和技术的杰作，也是非常令人愉快的事情，且不说这些思想和技术还可能作为我们工作中的借鉴。

我们非常高兴地看到，本书的出版将为图书市场上数量极少的类似书籍增添新的一员，也使关心这一领域的读者，准备和正在开设这方面课程的教师，学习这方面课程的学生增加一种新的选择。本书的翻译工作由我负责组织，我的一些学生参予。前言和第1、2、8、9、10、11章由我翻译，第3、4章由蒲戈光翻译，第5章由魏晋伟翻译，第6、7章由林宗芳翻译，第12、13、14、15章由秦胜潮翻译，最后由我统一修改定稿，秦胜潮也参与了一些修订。蒲戈光还为这个工作实现了一个术语对照查询的帮助系统。本书译稿的每一章都经过至少两个人多遍阅读和修订。我们希望这个工作的结果能够使读者满意。

裘宗燕

2001年7月于北京大学

前　言

本书的设计是面向有关程序设计语言的大学三、四年级课程，至少要求前面已经学过一门有关程序设计的引论课程。如果加上一些辅助性阅读材料，这本书也可以用于研究生课程。

本版有什么新变化

人们对语言的观点变化以及使用本书的反馈意见是本次修订的最根本动力。教师们希望能强调概念，但也要求用比较少的语言来阐述有关的概念。在这段时期内，Modula-2已逐渐淡出，而C++已占据了重要地位，被作为开发软件产品所用的程序设计语言。目前函数式程序设计语言的候选者还应该包括Standard ML、Haskell和Miranda。

这个新版本包括15章，比第1版多了3章。新的3章的内容分别是：

- 数据类型，如数组、记录和指针，占有一章。
- 另一章用Standard ML介绍函数式程序设计。
- 最后一章是有关语言的总结。

语言描述和语法问题移到了前面的第2章。

本书的组织

本书特别强调的是概念和它们协同工作的情况，而不是各种语言特征。因此讨论中将相关的概念放在一起，以便能够同步地给出有意思的例子和程序设计练习。对于一种语言，只是根据例子和练习的需要介绍足够多的东西，对所有有关语言的总结放在第15章。

第一部分：引言

第1章讨论程序设计语言所扮演的角色及其开发过程。这里介绍了本书提出的各种程序设计范型，包括命令式、面向对象、函数式和逻辑式的程序设计。

第2章讨论程序设计语言的描述问题，以便使这些东西能够应用到本书的其他部分。这一章的例子主要讨论表达式，描述表达式语法的方法可以用到语言的其他部分。

第二部分：命令式程序设计

命令式语言族在第3～5章讨论。术语“命令式”（imperative）来自于命令和动作，这种计算模型就是基于基础机器的一系列动作。

第3章的主题是控制流。结构化的控制结构，如while语句，能用于组织程序的控制流，形成以结构化语句为单位的程序设计，而不是以单个赋值为单位。学过主要关注命令式程序设计课程的学生一般都熟悉Pascal，因此，这一章不仅讨论了赋值和结构语句，还讨论了借助不变式考虑程序设计的问题。所举例子涉及各种基本的值，例如整数和数组等。

第4章讨论命令式程序设计中的数据。在Pascal和C出现之后，各种数据表示机制，如数组、记录和指针，已经趋于稳定。有关这些机制的讨论中也结合了它们在第6章和第7章中用来表

目　录

第五部分 其他范型