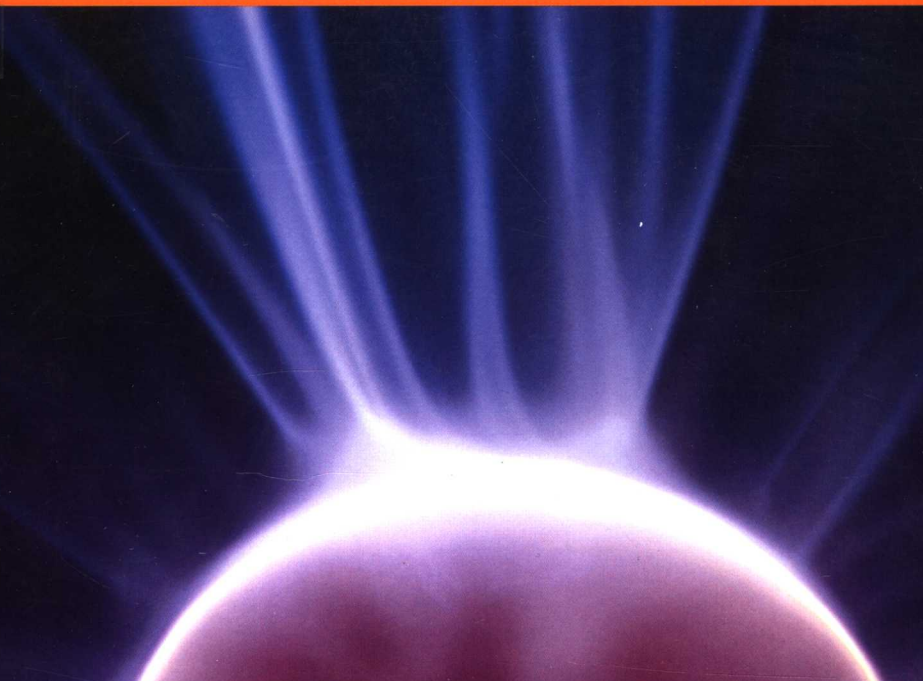




Concepts of Programming Language

(Sixth Edition)

程序设计语言概念

(第六版)

[美] Robert W. Sebesta 著
林琪 侯妍 译

国外经典计算机科学教材

TP312
1960

Concepts of Programming Language
(Sixth Edition)

程序设计语言概念

(第六版)

[美] Robert W. Sebesta 著
林琪 侯妍 译



中国电力出版社

www.infopower.com.cn

Concepts of Programming Languages, 6th edition (ISBN 0-321-19362-8)

Robert W. Sebesta

Copyright ©2003 by Pearson Education, Inc.

Original English Language Edition Published by Addison-Wesley.

All rights reserved.

Translation edition published by PEARSON EDUCATION ASIA LTD and CHINA ELECTRIC POWER PRESS, Copyright © 2006.

本书翻译版由 Pearson Education 授权中国电力出版社独家出版、发行。
未经出版者书面许可, 不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有 Pearson Education 防伪标签, 无标签者不得销售。

北京市版权局著作权合同登记号 图字: 01-2005-5996 号

图书在版编目 (CIP) 数据

程序设计语言概念/ (美) 西贝斯塔 (Sebesta, R.W.) 著; 林琪译. —北京: 中国电力出版社, 2004

ISBN 7-5083-2490-0

I.程... II.①西...②林... III.程序语言 IV.TP312

中国版本图书馆 CIP 数据核字 (2004) 第 075649 号

丛 书 名: 国外经典计算机科学教材系列

书 名: 程序设计语言概念 (第六版)

编 著: (美) Robert W. Sebesta

翻 译: 林琪 侯妍

出版发行: 中国电力出版社

地址: 北京市三里河路 6 号 邮政编码: 100044

电话: (010) 88515918 传 真: (010) 88518169

印 刷: 北京同江印刷厂

开 本: 185×233 1/16 印 张: 32.5 字 数: 799 千字

书 号: ISBN 7-5083-2490-0

版 次: 2006 年 1 月北京第 1 版 2006 年 1 月第 1 次印刷

定 价: 49.80 元

版权所有 翻印必究

出版说明

新世纪的朝阳刚刚露出丝抹微红,如火如荼的全球信息化浪潮便汹涌而至,让人无时无刻不感受到新一轮产业革命的气息。如何在这场变革中占尽先机,既是对民族信息业的挑战,也是机遇。从而,作为民族信息产业发展基石的高等教育事业就被赋予了比以往更重的责任,对培养和造就我国 21 世纪的一代新人提出了更高的要求。但在计算机科学突飞猛进的同时,专业教材的发展却严重滞后,越来越成为人才培养的瓶颈。同时,以美国为代表的西方国家计算机科学教育经历了充分的发展,产生了一批有着巨大影响力的经典教材,因此,以批判、借鉴的态度有选择地引进这些国外经典计算机教材,将促进国内教学体系和国外接轨,大大推动我国计算机教育事业的发展。

中国电力出版社进入计算机图书市场已有近 6 个年头,通过坚持“高端、精品、经典”战略,致力于与国外著名出版机构合作,出版了大批博得计算机业界和教育界赞誉的作品。通过与信息技术教育界人士的广泛沟通,同时依托丰富的出版资源,中国电力出版社适时推出了“国外经典计算机科学教材”的出版计划。本次教材出版计划是和美国最大的计算机教育出版机构——Pearson 教育集团(Addison-Wesley、Prentice-Hall 等皆为其下属子公司)合作,依托其数十年积累的大批经典教材资源,确保了教材选题的权威经典。

为保证这套教材的含金量,并做到有的放矢,我们在国内组织了由中国科学院、北京大学等一流院校教师组成的专家指导委员会,对高校课程教学体系做了系统、详细的调查,听取了众多教育专家、行业专家的意见,对教育部的教育规划进行了认真研究,并深入了解国外大学实际教学选用的教材状况,对国外教材做了理性的分析,确立了依托国家教育计划、传播先进教学理念、为培养符合社会需要的高素质创新型人才服务,来作为本次“国外经典计算机科学教材”出版计划的宗旨。

我们从 2002 年的下半年开始着手这套教材的策划工作,并多次组织了专家研讨会、座谈会等,分析现有教材的优点与不足,采其精华,并力争体现本套教材的质量和特色。

1. 深入理解国内的教学体系结构,并比照国外相同专业的课程设置,既具有现实的适用性,又立足发展眼光,具备一定的前瞻性。
2. 以计算机专业的核心课程为基础,同时配合专业教学计划,争取覆盖专业选修课程和专业任选课程。
3. 选取国外的最新教材版本,同时对照国内同专业课程的学时要求,对不适用的版本进行剔除,充分满足国内教学要求。
4. 根据专业对口和必须具备同课程教学经验的要求,严格挑选译者,并严把质量关,确保教材翻译的高质量。
5. 通过从原出版社网站下载勘误表及与原书作者进行沟通的方式,对原书中的错误一一做了修改。

6. 对教材出版的后期工作,如审校、编辑、排版、印刷进行了严格的质量把关。

经过专家指导委员会的集体讨论,并广泛听取广大高等院校师生的意见,反复比较,从数百种国外教材中遴选出数十种,列入第一阶段的出版计划。这些教材的作者无一不是学富五车的大师,如 Stallings, Date, Ullman, Aho, Bryant, Sedgewick 等,他们的作品均是一版再版,并被众多国外一流大学如 Stanford University, MIT, UC Bekerley, Carnegie Mellon Univeristy, University of Michigan 等采用为教材。拟订的第一阶段出版计划包括 30 种图书,内容覆盖程序设计、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、离散数学等计算机专业核心基础课程,基本满足国内计算机专业的教学要求。

此外,为了帮助广大任课教师加深对本系列教材的理解,减轻他们的备课难度,我们从国外出版机构引进了大批的课程教学辅助资料,并积极延请国内优秀教师,根据其使用该系列教材中的教学经验,着手编写更加适合国内应用状况的教辅材料。

由于我们对国内高校计算机教育存在认识深度上的不足,在选题、翻译、编辑加工出版等方面的工作中还有许多有待提高之处,恳请广大师生和读者提出批评和建议,并期待有更多的人加入到我们的工作中来。我们的联系方式是:

电子邮件: csbook@cepp.com.cn

联系电话: 010-88515918-300

联系地址: 北京市西城区三里河路 6 号中国电力出版社

邮政编码: 100044

前 言

第 6 版调整内容

《程序设计语言概念》(第 6 版)的主旨、整体结构以及表述方法与前 5 版如出一辙。本书的根本目的是向读者提供一些必要的工具,以便对现有以及将来出现的程序设计语言作出适当的评价(evaluation)。另外还力图帮助读者打好学习编译器设计的基础。之所以要在第 6 版中作出一些调整,其原因是多方面的。首先,为了使本书的内容顺应当前的发展,大量有关早期程序设计语言的讨论在此均被删去,取而代之的是一些更新的语言。不过,如若某些较早的程序设计语言具有特别有趣的历史,其相关内容则予以保留,分别放在“历史点评”中加以介绍。其次,如今学生在着手学习本课程时对于面向对象编程往往已有基本的了解,为了反映这一点,我们对书中内容作了相应更新。在此缩减了对基本知识的讨论,而对高级主题的讨论则有所扩展。

第三,许多修改得益于一些评论人员的意见。例如,函数式程序设计语言的内容得到了重新组织和加强。而且我们在多数章的后面增加了“程序设计练习”一节,使得学生能够对书中所讲述的概念进行实践,从而使这些概念更具现实性,更有意义。本书中有一个新的补充,即一个与本书相关的网站,其中包含了一些小型的语言手册,提供了与学生的问答互动,另外还对一些程序设计项目作了介绍。最后,对于目前取得广泛应用的一些语言,书中还在多处呈现了对其设计者的人物访谈。这些都体现了语言发展的人性化的一面。

与以前版本不同,第 6 版特别作出了四点调整。首先,对实现子程序的内容进行了精简,这很大程度上是因为 Pascal 和 Modula-2 日益淡出舞台,另外,Ada 的使用范围也有所缩减,这些都导致了实现拥有静态作用域的嵌套子程序的重要性已经不再那么突出。所有有关 Pascal 的示例都采用 Ada 进行了重写。第二,扩充了第 14 章,从而覆盖了异常处理和事件处理的内容。由于交互式 Web 文档得到了广泛的使用,随之而来,对事件处理的关注以及事件处理的重要性也日益增长,本书中作出以上修改也正是源自于此。第三,删去了对 Smalltalk 的介绍,这是因为我们认为 Smalltalk 的语法细节与本书的内容无甚关系。第四,由于原有的一些程序设计语言日渐“衰老”,而一些新生代的程序设计语言蓬勃出现,这些都促使本书作出了大量重要的修改。这一版中对于 Modula-2、Pascal 和 ALGOL 等语言鲜有提及。另外,有关 Ada 和 Fortran 的内容也缩减为仅对它们的突出特点(在其他流行语言中不曾出现)进行介绍,而有关 JavaScript、PHP 和 C# 的新内容则在适当的地方予以补充。最后一点,当前版本的书中许多章都包括了新的一节,即“程序设计练习”。

本书内容

本书描述了程序设计语言的基本概念,在此讨论了不同语言构造(译者注:Construct 也称结构,本书中将此均译作构造,以区别于 Structure)的设计问题,分析了采用某些最常用的语言来设计这些构造的选择方法,并对诸多设计方案进行了仔细比较。

要对程序设计语言进行科学的研究,就必须对一些相关主题加以分析,其中包括描述程序设计语言语法和语义的形式化方法,相关内容将在第3章中介绍。对于不同语言构造的实现技术也必须考虑在内,词法和语法分析将在第4章讨论,子程序链接的实现是第10章的内容。另外一些语言构造的实现也将在本书中其他部分出现。

以下介绍是对第6版内容的概括。

章节组织

第1章首先从研究程序设计语言的基本原理入手,然后讨论了评价程序设计语言和语言构造的原则。另外,对语言设计的影响、常见的设计权衡以及基本实现方法也将在本章中介绍。

第2章概述了本书所涉及的大多数重要语言的发展历程。尽管这里并未对任何一种语言完整地加以描述,不过各种语言的起源、用途以及所做出的贡献在此将一览无余。这种对历史的追溯很有意义,对于理解当前语言设计的实践和理论基础来说,它将提供必要的背景知识,另外还可以促进对语言设计和评价的进一步研究。此外,由于书中的其余内容与第2章均无前因后果关系,因此可以独立于其他章节单独阅读。

第3章介绍了用于描述程序设计语言语法的主要形式化方法,即BNF(巴科斯范式)。其后介绍了属性文法,这是对语言的语法和静态语义的描述。接下来讨论的是语义描述,语义描述是一项很艰巨的工作,在此对三种最常用的方法作了简要介绍,即:操作语义、公理语义和指称语义。

第4章介绍了词法和语法分析。这一章所面向的是那些不再开设编译器设计课程的学校。与第2章类似,本章相对独立,可以不依赖于书中其他部分单独阅读。

第5~14章详细描述了有关命令式语言主要构造的设计问题。对于各种构造,均提供了针对多种示例语言的设计方法,并分别作出了评价分析。具体地,第5章介绍了变量的诸多特征,第6章涉及数据类型,第7章则解释了表达式和赋值语句。第8章描述了控制语句,第9章和第10章讨论了子程序及其实现。第11章介绍了数据抽象特性。第12章对支持面向对象程序设计的语言特性(继承和动态方法绑定)进行了深入探讨。第13章讨论了并发程序单元,第14章涉及异常处理和事件处理。

最后两章(第15章和第16章)描述了程序设计领域中另外两种最重要的选择:函数式程序设计和逻辑程序设计。第15章介绍了Scheme,包括其部分基本函数、特殊范式以及函数范式的描述,另外提供了一些用Scheme编写的简单函数的例子。在本章中还对COMMON LISP、ML和Haskell作了简要介绍,从而展示了多种不同类型的函数式语言。第16章所介绍的是逻辑程序设计和逻辑程序设计语言Prolog。

给授课教师的建议

在科罗拉多大学Colorado Springs本部,本书用于低年级学生的程序设计语言课程,课程安排如下:我们通常会详细介绍第1章和第3章,尽管学生们会发现第2章很有意思,而且读起来也很有收获,但是我们很少占用正课时间来介绍第2章,因为其中并没有太多难懂的技术内容。如前所述,由于后续章节中的内容均与第2章无关,因此这一章可以全部略去不讲,另外由于我们开设了一门有关

编译器设计的课程，因此第 4 章也被略去。

对于熟练使用 C++、Java 或 C# 的学生来说，第 5~9 章的内容相对容易，第 10~14 章的难度则要大一些，需要较详细地讲解。

对低年级的大多数学生而言，第 15 章和第 16 章介绍的是全新的内容。如果学生要学习这两章，最好为其提供 Scheme 和 Prolog 的语言处理器。在此提供了足够的资料，可以帮助学生尝试一些简单的程序。

作为本科生的课程，可能无法详细覆盖后两章的全部内容。不过，对于研究生课程，尽管可以略去有关命令式语言的部分章节，但非命令式语言（也称描述性语言）则应完整地加以讨论。

补充材料

读者可以在网站 www.aw.com/cssupport 上获得以下资料：

- 一组演讲文稿幻灯片。这些幻灯片都是 Microsoft PowerPoint 格式的源文件，各个幻灯片分别对应本书中的每一章。
- 书中所有图片的 PowerPoint 幻灯片，基于此你可以创建自己的演讲文稿。
- 一个相关网站。伴随第 6 版的推出，我们为学生们提供了一个全新的补充资源。为了巩固课堂上的学习，并对本课程的实践环节提供帮助，同时（或者）为了方便参加远程教学的学生，我们为第 6 版提供了一个相关网站，其内容如下：

(1) 关于某些语言的小型手册（大约 100 页的教程）。在此假设学生已经了解如何用另外某种语言进行编程，即掌握了足够的知识，从而能够顺利地通读有关各种语言的章节内容。目前已经提供了关于 C++、C、Java 和 C# 的手册。

(2) 实践项目。针对本书中所讲授的每个概念，在此提供了一系列实践项目。授课教师完全可以得到相应的解决方案。

(3) 自测练习。使用 Addison-Wesley 软件引擎，学生们可以完成一系列的多选和填空练习，从而检查自己对所读章节是否已确实掌握。

许多问题的解答只有得到授权的教师才能获得。要了解得到有关资源的信息，请咨询当地的 Addison-Wesley 销售代表，或者发 E-mail 至 aw.cse@aw.com。

语言处理器获取

对于本书中所讨论的部分程序设计语言，其处理器及相关信息可以在以下网站找到：

C#	http://microsoft.com
Java	http://java.sun.com
Haskell	http://haskell.org
Scheme	http://www.cs.rice.edu/CS/PLT/packages/drscheme/
Perl	http://www.perl.com

几乎所有浏览器均包括有 JavaScript，而 PHP 几乎在所有 Web 服务器中都能找到，在本书相关网站中也同样涵盖了以上信息。

致谢

第 6 版的诞生仰仗了多位杰出评论人员的贡献，他们的建议使这一版增色不少。在此以字母顺序将他们列出：

Charles Dana, 加利福尼亚州州立工业大学, San Luis Obispo

Eric Joanis, 多伦多大学

Donald H. Kraft, 路易斯安那州州立大学

Dennis L. Mumaugh, DePaul 大学

Sibylle Schupp, Rensselaer 工业学院

Neelam Soundarajan, 俄亥俄州州立大学

Ryan Stansifer, 佛罗里达理工学院

Steve Stevenson, Clemson 大学

Virginia Teller, Hunter College CUNY

Salih Yurtas, 德克萨斯州 A&M 大学

Yang Wang, 西南密苏里州州立大学

《程序设计语言概念》这本书发展至今，还有许多人在各个阶段为其之前的版本作出过贡献。他们的意见都很有见地，而且相当有建设性。同样在此以字母顺序将他们逐一列出：Vicki Allan、Henry Bauer、Carter Bays、Manuel E. Bermudez、Peter Brouwer、Margaret Burnett、Paosheng Chang、John Crenshaw、Barbara Ann Griem、Mary Lou Haag、Eileen Head、Ralph C. Hilzer、Hikyoo Koh、Jiang B. Liu、Meiliu Lu、Jon Mauney、Bruce R. Maxim、Robert McCoard、Michael G. Murphy、Andrew Oldroyd、Rebecca Parsons、Jeffery Popyack、Steven Rapkin、Hamilton Richard、Tom Sager、Joseph Schell、Mary Louise Soffa，以及 John M. Weiss。

还要向责任编辑 Maite Suarez-Rivas、策划编辑 Katherine Harutunian、Addison-Wesley 的执行总监 Juliet Silveri，以及 Argosy 的 Daniel Rausch 致以诚挚的谢意，正是他们的努力方使得第 6 版得以迅速问世，也是基于他们的帮助，才使得这一版较之于第 5 版更为完善。

最后，我要感谢我的孩子们，Jake 和 Darcie，为了撰写这本书的第 1 版直至第 6 版，我的分分秒秒似乎都花费于此而无法陪伴孩子们，感谢你们无尽的耐心。

关于作者

Robert Sebesta 是科罗拉多大学 Colorado Springs 本部计算机科学系副教授。Sebesta 教授于科罗拉多大学 Boulder 分部获得其应用数学学士学位，并在宾西法尼亚州州立大学获得了计算机科学硕士和博士学位。他在计算机科学领域执教长达 30 多年，所关注的专业领域是程序设计语言的设计和评价、编译器设计以及软件测试方法和工具。此外，他还是 ACM（美国计算机学会）和 IEEE 计算机学会的会员。

目 录

前 言

第 1 章 入门知识	1
1.1 为什么研究程序设计语言的概念	1
1.2 程序设计领域	3
1.3 语言评价标准	5
1.4 对语言设计的影响	14
1.5 语言分类	16
1.6 语言设计权衡	17
1.7 实现方法	18
1.8 编程环境	22
小结	23
复习思考题	23
有关问题集	24
第 2 章 主要程序设计语言的发展之路	25
2.1 Zuse 的 Plankalkül	26
2.2 最小硬件程序设计：伪代码	28
2.3 IBM 704 和 Fortran	29
2.4 函数式程序设计：LISP	34
2.5 向优化前进的第一步：ALGOL 60	38
2.6 计算机化的商业记录：COBOL	43
2.7 分时的起步：BASIC	47
人物访谈	50
2.8 面向全盘：PL/I	52
2.9 两种早期的动态语言：APL 和 SNOBOL	55
2.10 数据抽象的萌芽：SIMULA 67	56
2.11 正交设计：ALGOL 68	56
2.12 ALGOL 之后的一些早期语言	58
2.13 基于逻辑的程序设计：Prolog	63
2.14 历史上付出最大设计投入的语言：Ada	64
2.15 面向对象程序设计：Smalltalk	68
2.16 结合命令式和面向对象的特性：C++	71
2.17 基于命令式的面向对象语言：Java	73
2.18 面向 Web 的脚本语言：JavaScript 和 PHP	76
2.19 新千年的新语言：C#	79
小结	80

参考书目	80
复习思考题	81
有关问题集	82
第3章 描述语法和语义	84
3.1 引言	84
3.2 描述语法的一般问题	85
3.3 描述语法的形式化方法	86
3.4 属性文法	97
3.5 描述程序的含义：动态语义	102
小结	115
参考书目	116
复习思考题	116
有关问题集	117
第4章 词法和语法分析	120
4.1 引言	120
4.2 词法分析	121
4.3 解析问题	124
4.4 递归下行解析	126
4.5 自底向上解析	131
小结	136
复习思考题	137
有关问题集	138
程序设计练习	139
第5章 名、绑定、类型检查和作用域	140
5.1 引言	140
5.2 名	141
5.3 变量	143
5.4 绑定概念	145
人物访谈	148
5.5 类型检查	152
5.6 强类型	153
5.7 类型兼容性	154
5.8 作用域	156
5.9 作用域和生命期	163
5.10 引用环境	163
5.11 命名常量	165
5.12 变量初始化	166
小结	167
复习思考题	167
有关问题集	168
程序设计练习	173

第6章 数据类型	174
6.1 引言	174
6.2 基本数据类型	175
6.3 字符串类型	178
6.4 用户自定义序数类型	181
6.5 数组类型	184
6.6 关联数组	194
人物访谈	194
6.7 记录类型	197
6.8 联合类型	201
6.9 指针和引用类型	204
小结	213
参考书目	214
复习思考题	214
有关问题集	215
程序设计练习	216
第7章 表达式和赋值语句	217
7.1 引言	217
7.2 算术表达式	218
7.3 重载操作符	224
7.4 类型转换	225
7.5 关系和布尔表达式	228
7.6 短路计算	230
7.7 赋值语句	231
7.8 混合模式赋值	234
小结	234
复习思考题	234
有关问题集	235
程序设计练习	237
第8章 语句级控制结构	238
8.1 引言	238
8.2 选择语句	239
8.3 循环语句	246
人物访谈	253
8.4 无条件分支	256
8.5 卫哨命令	257
8.6 结论	260
小结	261
复习思考题	262
有关问题集	262
程序设计练习	262

第9章 子程序	265
9.1 引言	265
9.2 子程序基本知识	266
9.3 子程序的设计问题	270
9.4 局部引用环境	271
9.5 参数传递方法	272
人物访谈	277
9.6 子程序名作为参数	287
9.7 重载子程序	289
9.8 类属子程序	289
9.9 函数的设计问题	293
9.10 用户自定义重载操作符	294
9.11 协同程序	294
小结	297
复习思考题	297
有关问题集	298
程序设计练习	299
第10章 实现子程序	300
10.1 调用和返回的一般语义	300
10.2 实现“简单”子程序	301
10.3 实现有栈动态局部变量的子程序	303
10.4 嵌套子程序	308
人物访谈	309
10.5 块	315
10.6 实现动态作用域	317
小结	319
复习思考题	320
有关问题集	320
第11章 抽象数据类型和封装构造	323
11.1 抽象的概念	323
11.2 数据抽象引言	324
11.3 抽象数据类型的设计问题	326
11.4 语言示例	327
人物访谈	327
11.5 参数化抽象数据类型	337
11.6 封装构造	339
11.7 命名封装	342
小结	343
复习思考题	344
有关问题集	345
程序设计练习	345

第 12 章 对面向对象程序设计的支持	346
12.1 引言.....	346
12.2 面向对象程序设计.....	347
12.3 面向对象语言的设计问题.....	348
12.4 Smalltalk 中对面向对象程序设计的支持.....	351
12.5 C++中对面向对象程序设计的支持.....	353
人物访谈	353
12.6 Java 中对面向对象程序设计的支持.....	361
12.7 C#中对面向对象程序设计的支持.....	363
12.8 Ada 95 中对面向对象程序设计的支持.....	364
12.9 JavaScript 的对象模型.....	367
12.10 面向对象构造的实现.....	369
小结.....	372
复习思考题.....	373
有关问题集.....	374
程序设计练习.....	374
第 13 章 并发	375
13.1 引言.....	375
13.2 子程序级并发介绍.....	377
13.3 信号量.....	380
13.4 监视器.....	384
13.5 消息传递.....	388
13.6 Java 线程.....	398
13.7 C#线程.....	404
13.8 语句级并发.....	405
小结.....	407
参考书目.....	407
复习思考题.....	408
有关问题集.....	409
程序设计练习.....	409
第 14 章 异常处理和事件处理	410
14.1 异常处理介绍.....	410
14.2 Ada 中的异常处理.....	415
14.3 C++中的异常处理.....	420
14.4 Java 中的异常处理.....	423
人物访谈	424
14.5 事件处理介绍.....	430
14.6 Java 的事件处理.....	431
小结.....	437
参考书目.....	437
复习思考题.....	438

有关问题集	438
程序设计练习	440
第 15 章 函数式程序设计语言	441
15.1 引言	441
15.2 数学函数	442
15.3 函数式程序设计语言的基础知识	444
15.4 第一个函数式程序设计语言: LISP	444
15.5 Scheme 介绍	447
15.6 COMMON LISP	459
15.7 ML	460
15.8 Haskell	462
15.9 函数式语言的应用	465
15.10 函数式和命令式语言的比较	466
小结	466
参考书目	467
复习思考题	467
有关问题集	467
程序设计练习	468
第 16 章 逻辑程序设计语言	470
16.1 引言	470
16.2 谓词算子的简要介绍	471
16.3 谓词算子和证明定理	473
16.4 逻辑程序设计概述	475
16.5 Prolog 的起源	476
16.6 Prolog 的基本元素	476
16.7 Prolog 的缺陷	487
16.8 逻辑程序设计的应用	491
小结	492
参考书目	493
复习思考题	493
有关问题集	493
程序设计练习	494
参考文献	495

第 1 章 入门知识

本章内容:

- 1.1 为什么研究程序设计语言的概念
- 1.2 程序设计领域
- 1.3 语言评价标准
- 1.4 对语言设计的影响
- 1.5 语言分类
- 1.6 语言设计权衡
- 1.7 实现方法
- 1.8 编程环境

在着手讨论程序设计语言的概念之前，需要先了解一些入门知识。作为计算机科学专业的学生和专业软件开发人员，为什么需要研究语言设计和评价的一般概念呢？首先我们将对此作出一些解释。如果有人误认为拥有一到两种语言的实践经验就堪称计算机专家，那么上述讨论将使你获益匪浅。其后将简要描述主要的程序设计领域。接下来，由于本书将对语言构造和特性进行评价，所以在此提供了一组标准，我们将据此作出判断。其后所讨论的是对语言设计的两大主要影响，即机器体系结构和程序设计方法论。并介绍了在语言设计过程中需要考虑在内的一些主要权衡。

由于本书涵盖了程序设计语言实现的内容，本章还将对最常用的实现方法提供概述。最后，简要地描述一些编程环境的例子，并讨论它们对软件生产的影响。

1.1 为什么研究程序设计语言的概念

毋庸置疑，学生们很可能不清楚通过研究程序设计语言概念能够有多大收获。毕竟，计算机科学领域中还有许多其他主题值得仔细研究。以下列出了研究语言概念可能带来的好处，我们认为这足以令人信服。

- **提高表达思想的能力。**一般认为，在进行思想交流时，所用语言的表述能力对于我们考虑问题的深度会产生很大影响。如果某人不能流畅地操纵自然语言，其思想势必相对简单，特别是抽象能力会非常单薄。换句话说，人们很难对其无法描述的结构建立概念，这里的描述可以是口头上的，也可以是书面的。程序员在开发软件过程中也同样会受此局限。他们开发软件所用的语言会对其能够使用的控制结构、数据类型以及抽象的种类有所限制；相应地，所能构造的算法也会受限于此。

如果把目光放宽，了解到还有更多的程序设计语言特性，就会使软件开发中的这种限制有

所减弱。程序员可以通过学习新的语言构造拓宽其软件开发思路。

有人也许会提出异议,认为学习其他语言的功能对于以下情况可能于事无补,即,如果要求程序员只能使用某种语言,而这种语言并无上述功能,那么新功能的学习就无甚帮助。不过,这种想法是不成立的,这是因为,尽管一些语言可能没有对某些语言构造直接提供支持,但通常可以用这些语言进行模拟。

例如,如果一个C程序员了解了Perl(Wall等,2000)中关联数组的结构和使用,他就可以用C语言设计结构,从而模拟Perl中的关联数组。

对程序设计语言概念的研究可以使我们尽享各种颇有价值的语言特性,并且能够鼓励程序员使用这些特性。

尽管语言的许多特性都可以用其他语言来模拟,但设计出拥有最优特性集的语言仍然相当重要,而且其重要性并不会因为特性可模拟而有所减弱。最好直接使用一种特性(即该特性的设计集成在语言中,与之成为一体),而不是使用该特性的某种替代品,这种替代品往往不够精巧,会相对笨拙一些,而且在不支持该特性的语言中加以使用通常也不太安全。

- **改善选择适当语言的背景条件。**许多专业程序员并非计算机专业科班出身;相反,他们可能是自学编程,或者是通过内部培训班上路的。这些培训班通常只教授一种或两种与当前项目直接相关的语言。还有许多程序员尽管接受了正规的培训,但年月已久,他们所学的语言已经不再使用,而对目前在程序设计语言中常用的许多特性却知之甚少。这种背景条件会导致一个直接后果,即许多程序员在面对一个新的项目时,尽管可以从多种语言中作出选择,但是仍然固执地沿用其最熟悉的那种语言,即便这种语言并不胜任新的项目也不忍舍弃。如果这些程序员还熟悉其他一些可用的语言,特别是了解那些语言中的特殊特性,那么他们在作出语言选择时将会更加游刃有余。
- **增强学习新语言的能力。**计算机程序设计仍然是一个相对较新的学科,而设计方法论、软件开发工具以及程序设计语言也仍在不断取得进展。这就使得软件开发成为一个颇具吸引力的行业,但这也说明我们必须“活到老,学到老”。要学习一种新的程序设计语言,其过程可能是冗长而困难的,特别是对于那些只了解一、两种语言而从未接触过程序设计语言一般概念的人来说更是如此。一旦对语言的基本概念有了一个整体的认识,再来看这些概念是如何结合到所学语言的设计中会容易得多。

例如,对于理解面向对象程序设计概念的程序员,他们在学习Java(Gosling等,1996)时就会比那些从未使用过这些概念的人轻松很多。在自然语言的学习中也存在这种现象。你对母语文法掌握得越透,就越容易学习第二种语言。不仅如此,学习第二种语言还会对第一种语言的掌握起到很好的促进作用。

最后一点,具体做开发工作的程序员应当了解程序设计语言的用词和基本概念,以便阅读和理解程序设计语言手册以及有关语言和编译器的更高深文献。

- **更好地理解实现的重要性。**在学习程序设计语言的概念时,要对影响这些概念的实现问题有所关注,这不仅很有意思,同时也是必要的。在有些情况下,如果理解了实现问题,则会使我们对于语言之所以如此设计有所理解。这会进一步促使我们更聪明地使用语言(这正是语言的设计初衷)。通过理解如何在程序设计语言构造中作出选择,以及由这些选择所带来的结果,我们可以成为更棒的程序员。