国外计算机科学教材系列

程序设计语言

设计与实现(第3版)

Programming Languages

Design and Implementation (Third Edition)

Terrence W. Pratt, Marvin V. Zelkowitz 著

傅育熙 黄林鹏 张冬茉 等译校

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

PRENTICE HALL 出版公司

国外计算机科学教材系列

程序设计语言：设计与实现
（第3版）

Programming Languages
Design and Implementation（Third Edition）

Terrence W. Pratt, Marvin V. Zelkowitz 著

傅育熙 黄林鹏 张冬茉 等 译校

PRENTICE HALL 出版公司

電子工業出版社

内 容 提 要

本书在一定的广度和深度上为学生介绍了程序设计语言的主要概念，以提高他们的编程能力。全书分为两部分：第一部分讲述语言设计与实现的基本概念、数据对象和数据类型、子程序、顺序控制和数据控制；第二部分包括过程式语言（Fortran77、Pascal）、面向对象语言（C++）、函数式语言（Lisp）及逻辑式语言（Prolog）等内容。

本书可用作计算机系本科生的教材或参考书，也可供计算机软件开发人员参考。

丛 书 名： 国外计算机科学教材系列
原 书 名： Programming Languages: Design and Implementation (Third Edition)
书　　名： **程序设计语言：设计与实现（第 3 版）**
著　　者： Terrence W. Pratt, Marvin V. Zelkowitz
译 校 者： 傅育熙 黄林鹏 张冬茉 等 译校
责任编辑： 范宜清 陈　斌
印 刷 者：北京市天竺颖华印刷厂
出版发行：电子工业出版社出版、发行　URL：http://www.phei.com.cn
北京市海淀区万寿路 173 信箱　邮编　100036　发行部电话 68214070
经　　销：各地新华书店经销
开　　本：787×1092 1/16　印张：30.25　字数：730 千字
版　　次：1998 年 11 月第 1 版　1998 年 11 月第 1 次印刷
印　　数：7000 册
书　　号：ISBN 7-5053-4599-0/TP·2180
定　　价：45.00 元
著作权合同登记号　图字：01-97-1863

出版说明

计算机科学的迅速发展是20世纪科学发展史上最伟大的事件之一。从1946年第一台笨重而体积庞大的计算机的发明至今，仅仅半个多世纪，计算机已经变得小巧无比却又能力非凡。它的应用已经渗透到了社会的各个方面，成为当今所谓的信息社会的最显著的特征。

处于世纪之交科技进步的大潮中，我国正在加强计算机科学的高等教育，着眼于为下一世纪培养高素质的计算机人才，以适应信息社会加速度发展的需要。当前，全国各类高等院校已经或计划在各专业基础课程规划中增加计算机科学的课程内容，而作为与计算机科学密切相关的计算机、通信、信息等专业，更是在酝酿着教学的全面革新，以期规划出一整套面向21世纪的、具有中国高校计算机教育特色的课程计划和教材体系。值此，我们不妨借鉴并引进国外具有先进性、实用性和权威性的大学计算机教材，洋为中用，以更好地服务于国内的高校教育。

美国Prentice Hall出版公司是享誉世界的高校教材出版商，自1913年公司成立以来，即致力于教育图书的出版。它所出版的计算机教材在美国为众多大学所采用，其中有不少是专业领域中的经典名著。许多蜚声世界的教授学者成为该公司的资深作者，如：道格拉斯·科默(Douglas Comer)，安德鲁·坦尼伯姆(Andrew Tanenbaum)，威廉·斯大林(William Stallings)……几十年来，他们的著作教育了一批批不同肤色的莘莘学子，使这些教材同时也成为全人类的共同财富。

为了保证本系列教材翻译出版的质量，电子工业出版社和Prentice Hall出版公司共同约请北京地区的清华大学、北京大学、北京航空航天大学，上海地区的上海交通大学、复旦大学，南京地区的南京大学、解放军通信工程学院等全国著名的高等院校的教学第一线的几十位教师参加翻译工作。这中间有正在讲授同类教材的年轻教师和博士，有积累了几十年教学经验的教授和博士生导师，还有我国著名的计算机科学家。他们的辛勤劳动保证了本系列丛书得以高质量地出版面世。

如此大规模地引进计算机科学系列教材，在我们还是第一次。除缺乏经验之外，还由于我们对计算机科学的发展，对中国高校计算机教育特点认识的不足，致使在选题确定、翻译、出版等工作中，肯定存在许多遗憾和不足之处，恳请广大师生和其他读者提出批评、建议。

电子工业出版社
URL：http://www.phei.com.cn
Prentice Hall出版公司
URL：http://www.prenhall.com

序

《程序设计语言：设计与实现（第3版）》（Pratt 著）一书，以其内容翔实，论述精辟而著称。80年代中期，我访问美国各大学时，当时多数学校均采用它作为研究生教材。随着计算机科学的发展，作者在内容和形式等方面又对本书加以更新、充实。该书已有多种版本出版，我们目前翻译的是作者在1996年出版的最新版本。

本书可以从纵、横两方面加以考察。在纵向方面，介绍了各种语言的构造成分、结构、演变过程及其相应实现技术等方面的内容。在横向方面，则以语言设计中的一些普遍问题对各种语言进行比较评价，指出其优缺点。80年代曾经涌现出新一代的程序设计语言和程序设计风格，如函数程序设计、逻辑程序设计和面向对象程序设计，本书均辟有专章加以介绍。

我曾有意于该书的翻译，由于种种原因未能实现。我系傅育熙、黄林鹏、张冬茉等诸位同仁于工作之余，翻译了此书。他们作了许多有益的工作，他们的工作热情和敬业精神令人钦佩。由于时间仓促，加以经验缺乏，错误疏漏之处在所难免，望海内外同仁不吝指教。

本书可作为计算机专业本科生、研究生教材，也可供计算机专业工作者参考。

孙永强

于上海交通大学

译者的话

《程序设计语言:设计与实现(第3版)》是一本关于程序设计语言理论和实践的经典著作和教学用书,是任何一个想从事程序设计语言设计的计算机科学工作者的必读之书,也是一部程序设计语言发展史和程序设计语言工作原理的优秀教材。

该书从软件和体系结构两方面叙述了程序设计语言的各个侧面,从效率和正确性方面探讨了程序设计语言的发展方向,该书分为两部分:

第一部分讨论程序设计语言的基本概念,包括程序设计语言发展史、程序设计语言的设计准则、语言的翻译、数据类型、抽象、顺序控制、子程序结构、继承和程序设计语言发展方向等问题。

第二部分讨论了一些在程序设计语言发展史上占有重要地位而如今仍有着强大生命力的语言,涉及过程式语言、面向对象语言、函数式语言和逻辑式语言,包括 FORTRAN、C、Pascal、Ada、C++ 、Smalltalk、LISP、ML 和 Prolog 等,作者分别就历史背景、数据结构、控制机制、子程序、存储管理、抽象和封装等方面展开讨论,对于每种语言作者都给出了两个包含注解的完整例子,还提供了进一步阅读的资料清单。对于当前的一些语言热点,如 Java、HTML 等,作者也作了详细的分析和讨论,并把它作为教师指导用书的一章。

我们相信译本的出版对国内计算机科学教学质量的提高和程序设计语言研究的发展具有重要的意义。

本书的翻译历时两年,分为两个阶段,第一个阶段由上海交通大学计算机科学与工程系的程序设计研究室负责,其中傅育熙教授和陈伟杰同志翻译了第 1 章,黄林鹏副教授翻译了 2、13、14 三章, 王绍英老师翻译了第 3 章,进修学者姜利翻译了第 5、11、12 章,博士后袁伟翻译了第 7 章并和博士生曾小平一起翻译了第 4 章,博士生董宏和杨继锋分别翻译了 6、8、9 三章。此后我们在《程序设计语言概论》和《程序设计方法学》两门课程中试讲了该书的部分内容。1998 年夏季我们对以前的译稿进行整理校对,并重新翻译了该书一些章节,其中傅育熙校译了 1、6、7、8 四章,黄林鹏校译了 2、9、10.1、11 、12 、13、14 等七章,张冬茉校译了 3 、4、5、10.2 等章节。

译者感谢上海交通大学候文永教授对译者的支持及提供的帮助,感谢孙永强教授对译者的鼓励及为本书所写的序言,感谢陈斌同志对译文所提的宝贵意见,最后还要感谢邓又强先生的鼓励,没有他的努力就不会有本书的问世。

由于译者专业水平和驾驭中英文的能力有限,译文可能存在一些不妥之处,对此我们表示歉意并欢迎读者不吝指教。

译者

1998 年 8 月于上海

前　言

本书是《程序设计语言：设计与实现》的第三版。在前两版中，我们基于软、硬件结构基础——它们正是用那些语言写的程序运行所必需的——描述了程序语言的设计。第三版将沿袭这一传统，以帮助程序员开发出既正确又能高效执行的软件。在这一版本中，我们将继续这一做法，并且加强了关于构成理论和形式模型的基础的介绍，因为正是理论和模型构成了那些语言的基础。

在计算机界，程序语言设计——即语言的“诞生”、“成熟”和最终的“消亡”——仍是一个十分活跃的分支。在第三版中，我们详细介绍了九十年代中期的主流语言。有关 COBOL、PL/I、SNOBOL4 和 APL 的章节已经删去。增加的有关 C、C++、ML、Prolog 和 Smalltalk 的讨论，以反映程序语言设计的进化和社会上新范例的出现。Pascal 开始过时，而 Ada 和 FORTRAN 已经用新标准分别更新为 Ada 95 和 FORTRAN 90。猜测一下在本书的将来版本中哪些语言会如本版中的那样被收录也许很有意义。对我们来说，SNOBOL4 的删除是可观的损失。它曾经被发展为最有趣和最强大的语言，而现在则作为 PC 机上的共享软件而存在。

在马里兰大学，一门与本书结构一致的课程在过去的二十年中一直活跃在课堂。在我们的初级课程中，我们认为学生们已经从早期的课程中了解了 Pascal 和 C。然后我们强调 ML、Prolog、C++ 和 LISP，并包括在 C 与 Pascal 的实现领域的更深入的讨论。C++ 的学习将会加强学生们通过附加的面向对象的类得到的过程性语言的认识。本书并且包含 LISP、Prolog 和 ML 等不同的程序范例以供探讨。把这些语言中的一到两个替换为 FORTRAN、Ada 或 Smalltalk 也是可行的。

我们假设读者至少熟悉一种过程性语言，如标准 C、FORTRAN 或 Pascal。对那些使用本书作为初级教材的院校或那些希望籍此作为大型机编程语言设计的预备知识的人而言，第 1 和第 2 章提供了理解以后各章所必需的回顾性材料。第 1 章概括地介绍了程序语言，而第 2 章则是程序语言要求的简单概览。

本书的重心在程序语言的设计与实现领域。第一部分构成了大学程序语言课程的核心。第 3 章到第 8 章通过对以下这些语言设计的中心概念的描述构成了本课程的基础，这些概念是：程序语言语法模型的理论基础及其编译器（第 3 章）、基本元素类型（第 4 章）、封装（第 5 章）、语句（第 6 章）、过程援引（第 7 章）和继承（第 8 章），这些特征的实例将用不同的语言描述，而且还将讨论一些典型的实现策略。

本书的论题将包括 1991 年 ACM/IEEE Computer Society Joint Curriculum Task Force 就程序语言专题领域推荐的 12 个知识点〔TUCKER etal. 1991〕。对那些使用本书作为高级教材或想找到一些更高级主题的人来说，第 9 章继续了于第 3 章初次介绍的语法解析方面的讨论，并将通过对程序语言更新、外延语义及由非确定性和 NP 完备性介绍引出的演算方面的讨论带来程序语言语义的概念。这将为读者提供计算机科学中程序语言、软件工程和计算理论方面更高级课程的概览。对于这一部分的内容，谓词演算和数理逻辑方面的经验将大有帮助。另外，第 9 章还提出了并行计算领域的问题，提供了在软、硬件方面并行研究的介绍，还对将来程序语言设计领域的发展提出了建议。

编写编译器曾是计算机科学界中心课程，现在越来越多的人认为并非每一个计算机专业学生都需要具备开发编译器的能力，这项技术应该留给编译器设计专家。而删除了这项课程在课程表上留下的“空白”，也许可以用以下这些课程来更好地填补，如软件工程，数据库工程或其他一些实用的计算机科学技术。然而，我们坚信编译器设计技术对每一个优秀的程序员来说，都应成为他的背景知识中的一部分。因此，本书的一个焦点就是关注各种语言结构是如何编译的，而第 3 章将提供一个相当完整的语法解析方面的小结。

第一部分的九章强调了用 FORTRAN、Ada、C、Pascal、ML、LISP、Prolog、C++ 和 Smalltalk 编写的程序语言的实例。根据需要，还附加了 PL/I、SNOBOL4、APL、BASIC 和 COBOL 的例子。但是，第二部分则是按每个语言进行组织的。每一节均描述了一种不同的语言并显示它是如何体现第一部分九章所提出的特征的。其目的是表现每种语言是如何合理地实现本书前半部分所提出的软件结构的。尽管它们不是各种语言的使用手册，每一个章节还是为学生们提供了足够的信息以使他们不必一一查找各语言的使用手册就能使用这些语言来解决一些有趣的类问题(当然，如果你身边有这么一两本使用手册的话，对你的编程实现还是会有巨大帮助的)。

尽管在一个学期中简单地讨论一下所有这些语言也是可行的，但我们建议本课程的编程部分不应在上述每种语言中都包含那些问题。我们认为这将使课程流于肤浅。用九种不同的语言来写九段程序绝对是例行公事，而且这样提供给学生们的关于这些语言的知识也是没有深度的。我们认为教员应选择第二部分中三到四种语言加以详细介绍。

除了一些极为明显的例子外，本书中所有的实例均在正确的翻译器中测试过；但如同我们在 1.3.3 节中明确指出的那样，在我们的系统上的正确执行并不能够保证翻译器将按照标准语言来处理程序。我们相信某些“显然”的例子可能也有错误。如果确是如此，我们在此为可能出现的问题预先致歉。

综上所述，我们出版本书第三版的目标如下：

- 提供在现代编程语言发展中使用的关键范例的概览。
- 重点介绍某些提供所有特征的语言，用它们编写的程序在细节上足以体现那些特征。
- 详细深入地探索每一种语言的实现，以使程序员了解源程序及其实现行为之间的关系。
- 提供足够的形式理论以说明程序语言设计理论在一般计算机科学研究中的位置。
- 提供参考书和大量的习题以使学生有机会去拓展他们在这一重要问题上的知识。

对我们在本书的早期草稿中，由 Henry Bauer、Hikyoo Koh、John Mauney 和 Andrew Oldroyed 所加的极有价值的注释，以及马里兰大学 CMSC 330 的 118 位学生在 1995 年春季学期中所提供的改进了本书质量的有益反馈，我们谨此致以衷心的感谢。

在第二版与第三版间，有大约 70% 的段落已经被重写了。我们确信，本书的新版将在旧版上有相当可观的改进。希望您也同意这一点。

特里·普拉特

玛富·泽尔科韦兹

目　录

第一部分　概念

第二部分 范例和语言

第一部分　概　　念

在第一部分，我们将研究构成程序语言的基本特征。我们将讨论以下这些问题：构成程序语言的特征是什么？它们是如何相互作用的？又是如何实现的？描述程序执行的是哪些不同的范例？我们将用几种语言给出例子来作为对上述问题的解答。

在以后的第二部分，我们将独立地考察每一种语言并描述那些特殊的语言是如何体现上述问题的。

第 1 章　程序语言的学习

任何能描述算法及数据结构的标记可以构成一种程序语言；但是本书中我们的主要兴趣集中在那些在计算机上实现的语言。所谓语言的“实现”将在以下两章中加以考察。在第一部分的其他章节中，将详细考察一种语言各组成部分的设计与实现。其目的是研究语言的特征、一些语言的独立性并给出普遍使用语言的广泛的类的实例。

在本书的第二部分，我们将通过九种具体的语言及其实现来阐述这些概念的应用；这九种语言是：Ada、C、C++、FORTRAN、LISP、ML、Pascal、Prolog 和 Smalltalk。另外，我们也将对在本领域有影响的其他语言作简单的小结。它们包括：APL、BASIC、COBOL、Forth、PL/I 和 SNOBOL4。在开始正式的程序语言学习之前，还是值得探讨一下为何计算机程序员要进行这一学习。

1.1　为什么要学习程序语言

已经有数百种不同的程序语言被设计出来并实现了。甚至早在 1969 年，萨密特〔SAMMET 1969〕就列举了 120 种广泛使用的语言，而从那以后，又有许多语言产生了。然而，绝大多数程序员从不冒险使用多种语言，不少人仅完全使用一两种来完成编程。事实上，程序员的电脑上实际安装使用的仅仅是必需的特殊语言如 C、Ada 或 FORTRAN。那么，研究了各种各样不同但又可能永远用不上的语言又能使你得到什么呢？

只要你能透过对语言特征的肤浅认识并深入到设计概念的基础及语言实现的实效上，那将有许多理由进行这一学习。可以立即找到如下六个基本理由：

1. *加强你设计高效算法的能力*　许多语言都有这样的特点：当程序员能正确使用它们时，将十分有益；但如果运用有误，则会浪费大量的机器时间或将程序员引入耗时的逻辑错误中。即使是一位运用某一程序语言已有多年经验的程序员也不能保证他已经理解了该语言的全部特征。递归就是一个典型的例子，当正确地运用这个简单的编程技巧时，它会允许精致而高效的算法直接实现。但如果不能正确使用，它执行起