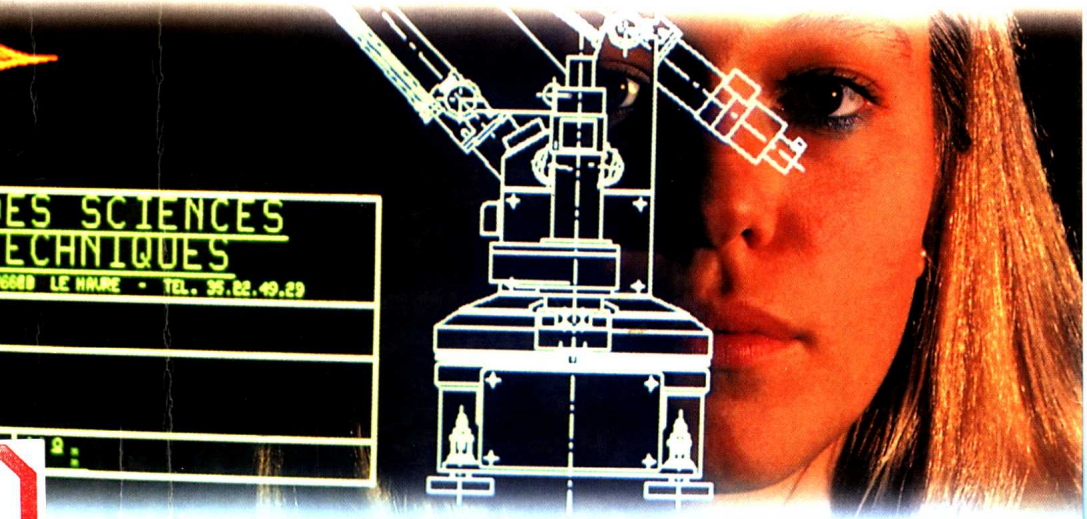
上海“九五”重点图书出版规划项目
全国名校计算机专业核心课程教材

程序设计语言原理

招兆铿　朱　洪　编著



上海科学技术文献出版社

上海“九五”重点图书出版规划项目
全国名校计算机专业核心课程教材

程序设计语言原理

招兆铿　朱　洪　　编著

上海科学技术文献出版社

责任编辑：方金善
封面设计：徐　利

程序设计语言原理
招兆铿　朱　洪　编著
*
上海科学技术文献出版社出版发行
(上海市武康路2号　邮政编码200031)
全国新华书店经销
上海教育学院印刷厂印刷
开本 850×1168　1/32　印张 16　字数 444 000
1998年7月第1版　1998年7月第1次印刷
印数：1－2000
ISBN 7－5439－1102－7/T・480
定　价：28.00元
《科技新书目》436－293

上海“九五”重点图书出版规划项目

全国名校计算机专业核心课程教材

编辑委员会

总　序

近20年来，计算机学科的发展，促使现代科学迅速崛起。当前计算机学科，已经成为实时控制、信息处理、通讯、企事业管理，以及社会生活各个方面无所不在，无所不用的必不可少的实用工具。计算机技术的应用，冲破了传统学科的分类，例如，经济、艺术、法律、农林、医学等各种学科，都需依赖于计算机的应用，除了各自领域的专业实施外，应用计算机已是各个专业提高效率，发挥潜能，促进发展，与专业息息相关的有效手段。同时，计算机应用的拓广，使计算机科学与现代科学技术相互结合，千丝万缕。当前，计算机专业教育中面临的最大问题是教学内容，远远滞后于应用实践。当前，个人电脑已深入到各行各业，网络，多媒体等在各方面应用普遍，但是在计算机专业教育中，这些内容还难成主流，而一些基础课程，如数据结构、数据库，以及计算机原理与组成等都还不能涉及前沿，有些课程内容，长期滞留在七、八十年代的水平。此外，计算机发展，已渗透到很多相关学科，如通讯、视像、激光、生物、化学、管理等各种相关领域，计算机教学的内容颇具知识爆炸之势，因此如何去芜存精，深厉浅揭是当前计算机教育改革的当务之急。本着上述观点，近年来，国内一些院校，曾对部分课程作了分析、实验、论证。大家认为当前世界计算机发展过程中，存在四大流行趋势，即是，面向对象、并行与分布处理、多媒体结构以及网络运行。如何在教学内容中，反映这四方面的应用是刻不

容缓，势所必然之事。

为此，我们经过筹划，组织了这套“全国名校计算机专业核心课程教材”(简称“名校教材”)。“名校教材”，并非都是名家，名作，它只是反映了国内一些名校在教学改革中的一些思路、举措。例如，当前计算机高级语言是大学的最基础先行课程。自C语言以后，很多异军突起，先有C++面向对象，又有VB，VC方便简捷，各领一方；网络流行之后，又有JAVA风靡一时，面对基础语言课程，观点各异，仁智相见，难执一词，难成共识。为此有人建议，语言应自概论讲起，分析语法结构，掌握语言构成规律，读通语言文本，那么任何计算机高级语言，都可举一反三，触类旁通。这种以结构规律来学会应用的方法，就是以不变应万变，任其千变万化，万变不离其宗。这种抓住本质，适应瞬息变化的拓展学科方法，在计算机的专业教育中是极具典型和富于启迪的。本丛书中，还有些教材，力图反映当前计算机学科中的最新进展，如数据库、操作系统两书都在详述基础概念基础上分类介绍了面向对象，并行与分布技术，以及网络、多媒体等实施的内容，使读者既能建立深厚基础，又能高屋建瓴，接近最新领域。

“名校教材”，目前已列为上海“九五”重点图书出版规划项目，并得到国内著名院校的教授、专家支持。

这套丛书，立意是着重基础，反映导向，注重实践。希望每种教材，能有创意，能具共鸣，能被接受，能予推广。但是另一方面，我们也意识到，由于各校情况各异，作者观感不一，理解角度有所不同，所以对教材的选用和编著，还不易一致认同，我们只希望这套教材不落窠臼，在反映当前学科动向，促进学以致用等方面，能起到推波助

澜作用。希望有关院校能根据本身条件，积极使用，参与讨论，以使各书能够不断修改，日臻完善。

最后我们感谢中国高等学校计算机教育研究会对“名校教材”策划的各种支持，感谢上海科学技术文献出版社为本丛书的立题、报审、出版等所付出的艰辛和努力。

左孝凌

1997.7

前　言

目前世界上已有百种以上不同的程序设计语言，这是根据各种不同的应用和领域设计的。其实，同一个领域的程序设计语言一般也是大同小异的。各种程序设计语言既有多样性又存在大量的共同性。因此，人们只要掌握这些语言的共同的主要概念和原理，在熟练掌握一个程序设计语言基础上，就能够触类旁通，举一反三地用更加省力和省时的办法学习另一种语言，甚至还可以具备设计某种简单语言的能力。

我们学习第一个程序设计语言时，基本上学习了三个方面的内容：程序设计方法、简单的数据结构和算法以及该语言的特性和用法。我们除了要进一步学习算法和数据结构的课程外，还应该系统地学习程序设计语言的定义、设计语言的基本准则、语法和语义、数据类型和它在机器中的表示、程序走向的控制、若干种类型的语言（过程式、函数式、逻辑式、面向对象、并行和分布式语言）等内容的课程。

《程序设计语言原理》参照 ACM/IEEE-CS 91 教程中提出的需求编写，目的向计算机专业学生和有关人员，介绍程序设计语言的基本概念和原理，介绍几种有代表性语言的特点，并且培养学生对程序设计语言的分析和设计能力。

本书的主要内容包括程序设计语言的语法和语义、

数据类型和控制结构、面向对象语言、逻辑程序设计语言和函数程序设计语言、以及并行程序设计语言等五个部分，分成十一章。本书使用最简单的例子介绍上述这些概念。每一章上给出了相应的练习。

第一章介绍程序语言的概念、抽象方法、计算的模式、语言的定义以及语言的翻译和设计等概念。

第二章介绍语言的有效性，语言的一般性、正交性和统一性以及语言的设计标准。

第三章介绍程序设计语言的词法结构、上下文无关语法、BNF 和句法图、句法分析树和抽象分析树，语法的二义性、结合性和优先性、分析的方法和工具。

第四章介绍语言的属性、联编和语义等，符号表、存储分配、表达式的求值，以及程序设计语言形式语义等。

第五章介绍数据类型和类型的信息、简单类型、类型的构造操作，类型的等价、检查和转换等。

第六章介绍卫哨命令和条件、循环的各种形式、GOTO 的争论、过程和参数以及例外处理。

第七章介绍抽象数据类型的代数规格说明、具体语言的抽象数据类型、重载和多态、程序模块和分块编译、抽象数据类型方法中的问题。

第八章介绍软件的重用性和独立性，对象、类和方法，继承，动态联编，几种具体的面向对象语言，面向对象语言的设计和实现。

第九章介绍程序作为函数、过程式语言中的函数程序设计、具体的函数程序设计语言、函数程序设计的数学方法：递归函数和 Lambda 演算、函数语言的动态存储管

理等。

第十章介绍逻辑和逻辑程序、Horn 子句、归结和合一、Prolog 语言、逻辑程序设计中的问题。

第十一章介绍并行处理和程序设计语言、伪并行和协同例程、信号量、管程和消息传递方法以及非过程语言的并行方法。

有关语义介绍的第四章由朱洪编写，其他十章由招兆铿编写。本书的出版得到了左孝凌教授的热心支持，在此表示感谢。

因水平有限，本书难免存在错漏和不妥之处，望读者指正。

编　者

1997 年 12 月

目　录

第一章　程序设计语言的概念

几十年来，程序设计语言在设计和使用上积累了大量的经验。虽然，在程序设计语言的设计上仍然有不够清楚的地方，但是，它的基本原理和概念是计算机科学知识的一个基本部分。对于程序员和计算机科学家，学习这些原理同学习具体的程序设计语言譬如 C 或者 Pascal，是同等重要的。没有这些知识，不可能掌握基本的观点，不可能使用我们同计算机的通信方法以及对计算机和计算的思考方法，不可能考察有效的程序设计语言及其设计。

这本书不采用专门针对一种具体语言的方法，来介绍一般程序设计语言的主要原理。作为例子，本书涉及的语言有，Pascal，Modula-2，C，Fortran，Ada，Lisp 和 Prolog 等。但是，读者不必熟悉这些语言的全部，也可以掌握有关的概念。对读者来说，最好具有其中一种语言的经验，以及有关数据结构，算法和计算过程的一般知识。

我们在本章介绍程序设计语言的基本表示和一些基本概念，同时简单地说明语言翻译程序的作用，但在本书上不准备讨论建立语言翻译程序的技术。

1.1　程序设计语言

程序设计语言的一般说法是："这是一种将人们想要做的工作告诉计算机的语言"。然而，这种说法是不够完整的。40 年代以前的计算机是用"接线方法"编程的。这就是说，程序员靠改变计算机的内部接线来执行某项任务。这是一种人们同计算机通讯的有效方法，但很难说它是程序设计语言。

自从 Jhon Von Neumann 提出了，采用作为数据存储的一系列代码来控制中央处理部件动作的方法代替"接线方法"后，40 年

代在计算机设计上有了很大的改进。程序员很快就意识到，使用符号表示命令代码和存储单元的方法能够大大节省编程的工作，于是，汇编语言便应运而生。

但是，汇编语言因为依赖具体机器，抽象水平低，难编写和难理解，所以不是人们通常想象的程序设计语言，在本书中也不准备讨论(为了同本书中讨论的高级语言区别，有时称汇编语言为低级语言)。实际上，程序员很快就意识到，用高级抽象的办法，可以提高精确地编写和能在不同机器上容易理解的命令的能力。比如赋值、循环和选择等的标准结构一直在使用，而且同具体机器无关。这些结构应该使用简单的容易翻译成机器语言的标准短语表示。例如。在 Pascal 上，表示汇编命令

```
LDA #2
STA X
```

(把数值 2 赋值给 X 单元)的赋值语句是

```
X:=2
```

这样，程序便在一定程度上和机器无关，但是这样的语言仍然反映 Von Neumann 机器的体系结构，即有一个同时存储程序和数据的内存储区域，一个按顺序执行由内存储提供的命令的独立的中央处理部件。

多数新型程序设计语言仍然保留这种计算的处理机模型优点。随着抽象办法的改进和新型系统结构(特别是并行处理器)的发展，使得程序设计语言不必基于任何具体的计算模型或者机器，而只需一般地描述计算或者处理。

定义 程序设计语言是一种用适合于计算机和人们的阅读方式描述计算的记号系统。

下面讨论这个定义中的三个主要概念。

计算 计算通常由图灵(Turing)机的数学概念形式定义。图灵机是一种计算机，它的操作简单，但能获得很精确的描述。它具有足以执行计算机所能够执行计算的能力。大家知道，图灵机的效率虽然低但是有能力执行现代计算机的任何计算。事实上，公认的

Church 定理指出，要建立一个在能力上比图灵机强的机器，是不可能的。

这本书使用的计算概念不是很形式的。我们把计算看作为可以被计算机执行的任何过程。然而，这不是说计算就是简单的数学演算，譬如两个数的乘积或者一个数的对数的计算。其实，这里的计算是指全部的计算机操作，包括数据加工、文件处理以及信息的存储和检索等。在这个意义上，计算就成了计算机上各类处理的同义词。有时，一个程序设计语言可以针对具体一类处理而设计，譬如报表生成、图示、或者数据库维护。这种专用语言也可以表示更一般的计算，但是，在本书中我们将主要讨论通用处理上，不是专用处理设计上的通用语言。

机器可阅读性 适合于计算机阅读的语言是指：这种语言必须具有一种足够简单的结构以便作有效的翻译。这种语言没有任何部分依赖于具体机器，只要求对于一般的需求，能够在翻译的确定性和复杂性范围内得到精确的叙述。这就是说，首先，必需有一个用来翻译语言的算法，这是一个有穷的无二义的逐步过程。第二，这个算法的复杂性不能够太大，多数的程序设计语言能够在与程序长度成比例的时间范围内得到翻译。不然，计算机在语言翻译处理上所用的时间会比实际计算的时间长。倘若把程序设计语言的结构限制为所谓的上下文无关语言，而且所有的翻译都基于这种结构，一般说来，上述的适合于机器的阅读性会得到保证。

人们的可阅读性 和适合于机器的阅读性不同，适合于人们的阅读性不要求很精确的表示，也不太详细。它要求程序设计语言提供容易理解的，对计算机动作的抽象表示。其实对人们来说，不需要完全熟悉机器的主要细节。因此，要求程序设计语言是一种类自然语言（譬如类似英文或中文）。从而，程序员从它的语言本能出发，可以直接理解被描述的计算（当然也会引起严重的误解）。

随着程序长度的增大，适合于人们的阅读性会有新的要求。（目前有些程序的长度同长篇小说一样）。大型程序的可阅读性需要恰当的办法，降低在对程序作整体理解时所需细节的总量。例