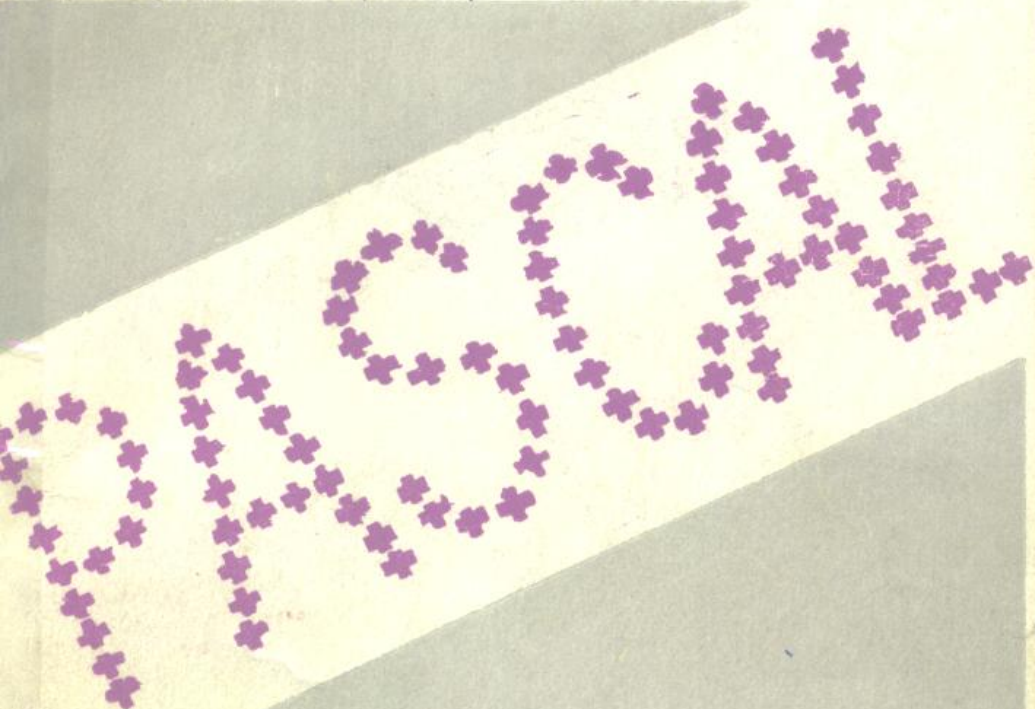


美J.E.·B.·考夫曼 著

嘉种 刘金宏 刘 峰 译



P A S C A L

解 题 及 结 构 程 序 设 计

中 国 铁 道 出 版 社

内 容 简 介

本书共分11章, 内容包括: 计算机组成与程序设计语言; 程序设计和解题的方法, PASCAL 语言的主要控制语句, 各种操作符、函数及标准型、纯量型和子界数据类型, 中级控制结构—自顶向下程序设计和函数; 结构型数据类型—数组及其操作; 记录型和集合型数组及其操作; 嵌套结构和递归, 大型程序系统的文件编制及调试技术; 分级分层记录文件和其他PASCAL 数据结构, 多维数组的操作, 指针变量和动态数据结构。

本书通过50个问题把解题和程序设计方法贯穿全书, 以分析问题、分步设计和由局部到整体的方法、实现 PASCAL 语言解题, 在每章中都有习题和作业, 书末附有答案。附录有: 终端分时系统的用法, PASCAL 语法图及UCSD PASCAL 字符串操作等。

15200/16
PROBLEM SOLVING AND STRUCTURED PROGRAMMING IN PASCAL

ELLIOT B. KOFFMAN

Addison-Wesley Publishing Company 1981

PASCAL 解题及结构程序设计

[美] E. B. KOFFMAN 著

丁嘉种 刘金宏 刘 峰 译

中国铁道出版社出版、发行

责任编辑 崔滨九 封面设计 安宏

各地新华书店经售

中国铁道出版社印刷厂印

开本: 787×1092毫米 1/32 印张: 19.5 字数: 440 千

1988年7月第1版 第1次印刷

印数: 0001—6,000 册 定价: 4.20 元

译 者 的 话

本书作者E·B·考夫曼是美国费城邓波尔(Temple)大学计算机与信息科学系教授。他曾与F·L·弗里德曼副教授合著《FORTRAN解题及结构程序设计》和《BASIC解题及结构程序设计》两本书。本书主要有以下特点:

第一,介绍的PASCAL语言的功能,同描述解题方法和程序设计技巧相结合,避免了学习程序设计语言枯燥的感觉,又可一举三得:学会了PASCAL语言;掌握了“自顶向下”、逐步求精的解题方法;训练了结构程序设计的技巧。

第二,选材与讲授方法着重循序渐进,逐步深化,并通过大量例题进行示范,引导学生养成良好的解题和程序设计的习惯。

第三,编排结构新颖。每章都扼要地对重点内容加以总结,并附有常见程序设计错误的说明。PASCAL语言的特性及程序设计风格都排印在显示表中,便于反复学习及经常查阅。

本书是大学PASCAL语言及程序设计课程较好的教科书和参考书,也是一本很好的自学读物。

本书第一至五章由刘金宏翻译,第六、七章由刘峰翻译,第八至十一章及附录部分由丁嘉种翻译。全书由丁嘉种审校。鉴于译者水平有限,译文不妥之处,请读者批评指正。

译 者

一九八六年

序 言

本书预定作为计算机程序设计入门课程的一学期教材。对于第一门课程学习别的语言的学生来说，它亦可用作学习第二门程序设计技术课程和 PASCAL 语言的教材。学习本书只需具有初等代数的基础知识。

本书的选材反映了这样的观点：应当在开发学生程序设计技能的最初期就引导他们养成良好的解题和程序设计习惯，并且最好是借助于例题、反复实践以及通过师生相互影响来逐步灌输给学生。解题和程序设计的技术通过求解各个应用领域里的五十多个问题进行示范。

解题方法和程序设计技巧的训练遍及全书。并且将一些比较新颖的教授法总结成一种独特的、结构优美的格式，这种格式在讨论每一问题时都重复出现。在这里，强调解题的三个基本阶段：分析问题；逐步详细说明算法；用 PASCAL 语言实现解题。

本书的首要目的是教授用逐步算法设计的技术来解题的方法。这种方法通过求解课文所讨论的每一问题反复进行说明。在描述解题所需的信息时强调使用“数据表”。鼓励学生在概述问题的解法时使用英语来描述每一个主要的算法步骤，然后以每一算法步骤为基础，再仔细地进行精细化。数据定义和算法精细化并行地加以描述；在算法步骤求精过程中所需的信息必要时就添加到数据表中去。

从一开始就着重指出：一个算法是作为一个控制语句序列来实施的。而且比较早地引入过程和函数模块的使用（第五章）。强调程序的模块化结构、并在全书采用之。

本书的编排使学生能较早地着手进行程序设计工作。课

文中每引入一个新的 PASCAL 功能特性, 就立即用于解例题, 而这些例题及其算法则清楚地说明了这些新功能。

书中详细地介绍了标准 PASCAL 语言的大部分功能特性, 并有单独的章节专门用于讨论高级的数据结构, 诸如文件、多维数组以及链表和树。还有一个附录讨论 UCSD PASCAL 中串的处理操作。给学生提供了足够多的例题和题解, 使他们对所有这些功能特性有一个透彻的了解。

书中内容作为一学期课程是绰绰有余的。前八章是全书的核心, 所有学生必须学习。后三章是关于文件、多维数组、链表和树方面的高深教材。这三章彼此无关, 可单独学习。因此, 教师可根据学生的兴趣以及可用学时的多少来选授这些章节。

每一章都附有可能出现的常见程序设计错误的说明以及调试方法的提示。大量的程序设计课外作业放在每章的末尾, 而练习题则插在正文之中。部分练习题的答案附在书末。

关于程序设计风格的讨论以醒目的显示表形式散列在全书各处, 书末列出其索引。

前四章介绍控制语句和数据结构 (IF、WHILE 和 FOR 语句——标准型和纯量型数据)。第五至八章介绍中级的课题 (函数、过程、参量、CASE 和 REPEAT——数组、记录 and 集合)。如前所述, 高级数据结构在第九至十一章讨论。

附录部分除字符串操作以及关于用穿孔卡片或终端时 PASCAL 的使用方法外, 还列出 PASCAL 保留字、专用标识符和运算符。另外, 还包括有全套的语法图。这些附录, 在教学和自学参考中都是有用的。

第一章先简要地讨论计算机的基本硬件、程序设计和程

序设计语言。然后介绍数据类型的概念以及变量和常量的使用。紧接着描述大多数计算机常用的一些基本操作。每一种操作及初级 PASCAL 语句表示,并说明每一种操作的功能。

在这里,使用初级 PASCAL 语句编写了一些简单的程序。使学生对 PASCAL 语言有一个初步的了解、对运行程序有足够的预备知识,以便他们能编写简短的程序并上机运行。PASCAL 程序使用穿孔卡片或终端时上机运行的方法在附录 2 和 3 里介绍。这些内容应当放在本章前几节中补充课文中的讨论。以便学会运行程序并获得直接的经验。

第二章阐明程序设计和解题的分步实现方法。数据表作为概括解题所需数据的工具加以介绍。同时详尽地讨论了算法的概念及其精细化,并且研究和解决判定和重复问题。本章最后介绍 IF 语句和 WHILE 循环结构。

第三章对 PASCAL 语言的 IF、WHILE 和 FOR 语句作较正规的描述,并给出一些程序范例。学完本章学生应会编写含有一个或几个判定的主要处理循环的程序。

第四章除详细讨论标准数据类型(整型、实型、布尔型和字符型)外,还讨论纯量型和子界型。全部运算符(包括 DIV 和 MOD)以及所有内部函数都在本章介绍。此外,还详尽地论述 EOLN 和 EOF 函数在读字符和数字数据时的使用方法。

第五章的主题是中级控制结构。将 CASE 语句和复合 IF 语句作为简化嵌套判定的手段介绍给读者。

本章还讨论分步(自顶向下)程序设计和程序模块化等重要概念。接着引入过程和函数,并在以后的章节中着重讨论它们的使用。此外,还定义变量的作用域,指出局部变量、非局部变量和全程变量之间的差别。

第六章介绍第一种结构型的数据类型：数组。透彻地讨论了数组类型以及具有各种不同下标类型和元素类型的数组例子。其中它包括对字符串数组的存贮和操作。

第七章描述记录型和集合型数据，介绍对记录进行操作的WITH语句。同样地，描述了集合操作的全部运算符，并举例说明它们的用法。

第八章涉及嵌套控制语句，较全面地介绍REPEAT语句和GOTO语句。讨论了大型程序系统的文件编制及其调试和测试技术。最后举例说明递归函数和递归过程的使用。

最后三章包括PASCAL的另一一些结构型的数据类型。可按任意次序学习这三章，由兴趣和可用学时数确定。第九章研究分级记录（记录中含有记录）和文件；第十章研究多维数组（数组中含有数组）；第十一章讨论指针、链表和树。

E·B考夫曼

1980年8月于费城

目 录

1. 计算机与程序设计导论	1
1.1 引 言	1
1.2 计算机组成	1
1.3 程序和程序设计语言	6
1.4 PASCAL语言简介	8
1.5 使用计算机	21
1.6 辅助的输入输出特性	21
1.7 数据类型简介	26
1.8 小 结	28
程序设计习题	29
2. 使用计算机解题	32
2.1 引 言	32
2.2 分析问题	32
2.3 描述解法	35
2.4 判定算法	40
2.5 循环算法	51
2.6 算法的实现	56
2.7 小 结	60
程序设计习题	62
3. 基本控制语句	64
3.1 控制语句简介	64
3.2 IF语句	65
3.3 WHILE语句	71
3.4 控制语句的应用	73
3.5 FOR语句	85

3.6 小机具(WIDGET)库存控制问题	92
3.7 调试程序与测试程序	98
3.8 常见的程序设计错误	105
3.9 小 结	106
程序设计习题	108
4. 标准型、纯量型和子界型数据	112
4.1 引 言	112
4.2 数值数据类型——实型和整型	114
4.3 算术表达式中的函数	123
4.4 布尔变量、表达式和操作符	129
4.5 字符型变量和函数	137
4.6 再论输入输出	143
4.7 纯量型和子界型数据	155
4.8 数值误差	166
4.9 常见的程序设计错误	167
4.10 小 结	170
程序设计习题	172
5. 中级控制结构	180
5.1 引 言	180
5.2 多分枝判定	180
5.3 自顶向下程序设计和函数	191
5.4 过程	203
5.5 自顶向下设计的应用	211
5.6 标识符的作用域	224
5.7 常见程序设计错误	232
5.8 小 结	233
程序设计习题	235
6. 数组和下标	246

6.1 引 言	246
6.2 数组说明	247
6.3 数组下标	251
6.4 数组元素的操作	254
6.5 整个数组的操作	264
6.6 填入部分元素的数组	271
6.7 字 符 串	281
6.8 字符串数组	292
6.9 常见程序设计错误	303
6.10 小 结	304
程序设计习题	304
7. 记录和集合	313
7.1 引 言	313
7.2 记录的说明	313
7.3 记录的处理——WITH语句	315
7.4 记录型数组	327
7.5 集合数据类型	332
7.6 集合运算	333
7.7 记录型数组的检索	348
7.8 常见程序设计错误	362
7.9 小 结	363
程序设计习题	365
8. REPEAT和GOTO语句嵌套结构和递归	368
8.1 引 言	368
8.2 REPEAT-UNTIL循环	368
8.3 嵌套循环	374
8.4 数组排序	377
8.5 GOTO语句	383

8.6 较大型问题的解法	387
8.7 程序系统的测试	396
8.8 递 归	397
8.9 常见程序设计错误	406
8.10 小 结	407
程序设计习题	407
9. 分级记录 and 文件	415
9.1 引 言	415
9.2 分级记录	415
9.3 记录的变体	420
9.4 文 件	429
9.5 文件更新和归并	438
9.6 文本文件	456
9.7 文件缓冲区变量	459
9.8 常见程序设计错误	466
9.9 小 结	467
程序设计习题	468
10. 多维数组	469
10.1 引 言	469
10.2 多维数组的说明	469
10.3 多维数组的操作	471
10.4 房间调配	481
10.5 计算机美术——画方块字简介	492
10.6 常见程序设计错误	500
10.7 小 结	501
程序设计习题	501
11. 指针变量和动态数据结构	506
11.1 引 言	506

11.2	NEW语句和指针.....	506
11.3	建立链式数据结构	511
11.4	删除结点	518
11.5	插入结点	522
11.6	多重链表和树	528
11.7	常见程序设计错误	541
11.8	小 结	542
	程序设计习题	542
附 录		547
附录 1	保留字、标准标识符和运算符	547
附录 2	穿孔卡片批量操作系统的用法	550
附录 3	终端分时系统的用法	552
附录 4	PASCAL语法图	557
附录 5	UCSD PASCAL 的字符串操作	567
部分练习题答案		577

1. 计算机与程序设计导论

1.1 引言

本章介绍计算机的一般结构，并讨论与计算机通讯所使用的语言（程序设计语言）。我们将看到所有计算机都是由四个基本部分组成——内存贮器、中央处理机、输入设备和输出设备。我们还将学习信息在计算机内存贮器中如何表示和处理。

本章也将对程序设计语言作一简单介绍。还要讨论两种程序设计语言：机器语言和高级语言，进而介绍一种高级语言——PASCAL，同时举例说明如何使用 PASCAL 语言指明计算机的某些基本操作。

1.2 计算机组成

1.2.1 引言

计算机是表示和处理信息的工具。有各种各样的计算机，从手中的计算器到占用几个房间以至整座楼房的大型复杂的计算机系统，大小不一，种类繁多。不久前，电子计算机价格昂贵得仅能用于企事业管理和科学计算，而现在适用于家庭使用的个人计算机比比皆是。

一台计算机的体积和价格，一般取决于它在规定的单位时间内能完成的工作量。比较大型的昂贵的计算机具有同时执行很多作业的能力，从而提高了他们的工作效能。这种计算机还配置有较多的设备来执行各种专门功能。这一切都增强了计算机的性能也提高了计算机的价格。

1.2.2 计算机的组成部分

除了在价格、体积和性能诸方面有较大区别之外，现代计算机在很多方面是非常相似的。一般来讲，一台计算机由四部分组成，如图 1.1 所示（连接各设备的直线表示信息流可能的径路，箭头表示信息的流向）。

首先，将由计算机处理的信息通过输入设备送到计算机的主存贮器。主存贮器中的信息送到中央处理机处理，并将处理结果送回主存贮器。存于主存贮器中的信息可以通过适当的输出设备输出。计算机各

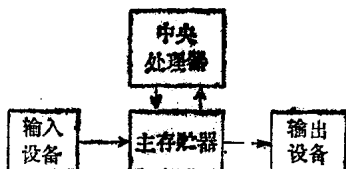


图1.1 计算机基本组成框图

组成部分及其相互作用在后面几节中还要进一步说明。

1.2.3 计算机的主存贮器

计算机主存贮器可以描绘成一个有序的内存单元序列。每一个单元都有其各自的地址。这个地址表示它在序列中的相对位置。图 1.2 表示一台有从 0 到 999 顺序编号的 1000 个内存单元的计算机主存贮器。某些大型计算机的主存贮器有数以百万计的内存单元。

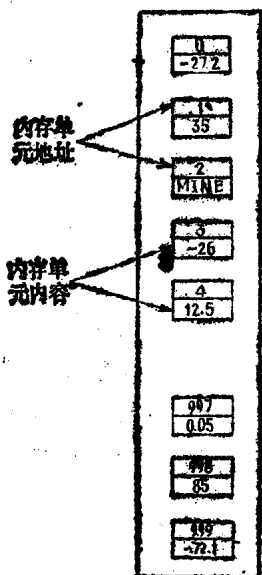


图1.2 具有1000个内存单元的计算机的内存贮器

计算机内存单元用于存放信息。各种信息——数字、姓名、表格、甚至图形都可以存贮到计算机内存中。某一内存

单元存贮的信息称作内存单元的内容。每一内存单元都存有信息，没有任何单元会是空的，而且也没有任何单元能同时存放一个以上的数据项。一个数据项一旦存入某一内存单元，它里面原来的信息就消失且不能自动恢复。图 1.2 中，3 号内存单元的内容是数 -26，4 号内存单元的内容是数 12.5。

重要的是要知道，处理数据的实际指令也存贮在计算机内存贮器中。这些指令由中央处理机加以解释并执行。

练习 1.1: 图 1.2 中 0, 2 和 997 号内存单元的内容是什么?

1.2.4 中央处理机

计算机存贮信息的能力对我们没有多大用处。真正有用的是计算机的处理能力。这种能力使人们能研究因计算上的要求而用其它工具不可能进行研究的问题。使用适当的指令，现代计算机能从原来的信息产生出大量新信息，解决许多其它工具不能解决的问题，并且使人们更深入了解这些问题的其它方面；计算机能在极其短暂的时间周期内完成这些工作。

中央处理机 (CPU) 堪称计算机解题能力的核心。CPU 能从内存中取出信息，这些信息可以是数据也可以是处理数据的指令。CPU 还能将处理结果送回内存以备后用。

CPU 中有一个算术——逻辑运算部件。算术运算部件由电子线路组成。它执行加、减、乘、除等各种算术运算。运算器可以在百万分之一秒数量级时间内执行这些运算。逻辑运算部件也由电子线路组成。它对信息进行比较并根据比较结果进行判断。正是计算机的这个特性再加上具有大容量的内存，使得它区别于目前我们许多人使用的简便的计算器。大多数这种计算器只能对数值数据进行算术运算，而不

能对数值进行比较、作出判断以及存贮大量数据。

1.2.5 输入输出设备

如果用户不能和计算机通讯,那么计算机的处理能力再强也没有多大用处。显然,必须能将信息输入到计算机内存,并能把存贮在计算机内存中的信息(通常指计算机处理的结果)取出来。输入设备就是用来将数据输入到计算机内存的装置。输出设备用于按易读的格式输出处理结果。

输入输出设备种类繁多。例如读卡机,纸带输入机和计算机终端均是输入设备。一种通常的输入设备是读卡机。这种设备读入的是一张张称为穿孔卡片的轻质卡。图1.3是一张穿孔卡片样品。穿孔卡可以穿到多达80列信息,其每一列是一个字符。图1.3的卡片中1到26列分别穿有字母A到Z,而36到45列分别穿有十进制数0到9。另外它上面还穿有+、-、·、,、(、)、*、/、\$、=和空格。每一字符表示成卡片上一列唯一的穿孔结构。一列上一个孔都不穿表示空格。

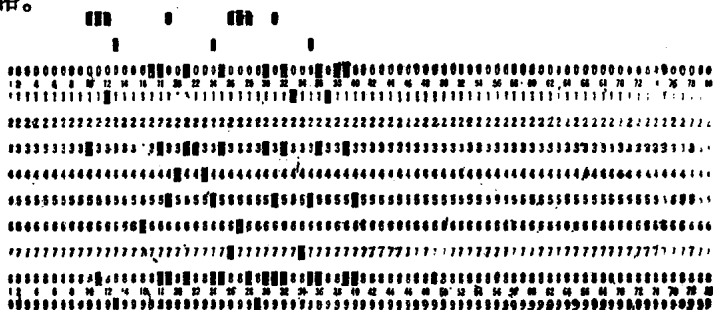


图1.3 穿孔卡片

待读入计算机的卡片在标准键控穿孔机上穿孔。键控穿孔机大多数穿孔键盘在穿孔的同时还在卡片的顶部打印出相应的字符。这种打印信息仅为人阅读卡片方便,它不能被读卡机读入。读卡机只读卡片上的孔。

一种通用的输出设备是行式打印机。这种设备通常一行打印120到132个字符,每分钟输出速度可达2000行以上。穿孔卡片也可以作为计算机的输出。第三种输出媒介是由称作绘图仪的设备,它将计算结果以图形来输出。到此我们所介绍的所有输出设备产生的都是计算结果的、永久性的和直接可读的记录。

计算机终端既用作输入设备也用作输出设备。终端通常由一个和打字机相似的键盘组成,计算机所需信息从键盘键入。计算结果以字母数字符的形式(字母和数字)打印在终端打印机的打印纸上或者显示在视频屏幕上。有些终端具有图象显示功能,它不仅能显示一行行字母和数字。而且可以输出二维图象。对于某些图象显示终端,用户可以使用一种称作光笔的电子指示器指向屏幕上显示的信息以便和计算机通讯。

计算机终端广泛应用于客票预定柜台的确认订票和打印客票工作。百货商店结帐柜台的计算机终端还用于为顾客记帐和货存控制。

在很多计算机系统中,另一种输入输出设备用于提供存取信息的辅助功能(辅助存贮)。这种设备可以在计算机内存与磁带、磁盘、磁鼓等磁性存贮介质之间传送大量信息。计算机在运行中可以取用事先存贮在辅助存贮器上的信息,而计算机新产生的信息也可以存入辅助存贮器以备今后检索使用。

设有两种存贮器的原因在于高速内存价格昂贵。辅助存贮器速度慢但价格低廉。它特别适用于存贮大量的不频繁使用的数据。为了对这些数据进行处理,必须先将其送入内存。