

梁秉岑 钟竞 孔俊宝 编著

计算机简明编程技术

——提高程序设计技
能的途径

兵器工业出版社

计算机简明编程技术

—提高程序设计技能的途径

梁秉岑 钟 竞 孔俊宝 编著

兵器工业出版社

内 容 简 介

本书是为学过一、二门计算机语言的学生，软件水平考试的应试人员以及从事计算机应用的科技工作者，进一步提高程序设计能力而编写的一本专著。它从计算机软件危机讲起，沿着程序的数据结构和控制结构两条脉络，论述了程序设计技术的成就及其发展。

全书共六章，第一章绪论；第二章结构程序语言PASCAL；第三章数据结构概览；第四章结构程序设计技术；第五章递归技术；第六章程序的正确性。

本书内容新颖，论述透彻，语言通俗，举例实用，是多年教学与研究的成果总结。很适合高等院校本科生、研究生作为教材，也可供技术人员阅读及自学。

计算机简明编程技术

——提高程序设计技能的途径

梁秉岑 钟 竞 孔俊宝 编著

卫洁 朱丽均 编辑

*

兵器工业出版社 出版发行

(北京市海淀区车道沟10号)

各地新华书店经销

三河印刷一分厂印装

*

开本 787×1092 1/32 印张：7 字数 150千字

1991年10月第1版 1991年10月第1次印刷

印数：00001—4500 定价 4.90元

ISBN 7-80038-350-4/TP·25

JS266/01

序

软件开发是一项具有高度技术水准的工作，在国外，它已吸引了一批极有聪明才智的计算机学者，并取得了重大的进展。但时至今日，人们开发优质软件的能力，仍然远远落后于计算机硬件发展和计算机应用的迫切要求。

软件工程是一门迅速发展的学科，它与其它工程科学和生产相结合，形成了一种新兴的科技产业——软件工业。在我国，软件工业很年轻，亟需培养大批熟悉程序设计技术，以及从事计算机应用科学研究的高级专门人才。

本书是作者们通过教学实践及实际工作中的应用而写的。本书的突出特点是既注重逻辑性和科学性，又注重实用性。原理性的论述，诸多程序设计技术的阐明，丰富而又饶有兴趣之典型实例的配合，是一本不可多得的程序设计技术的专著。

本书以程序结构（控制结构和数据结构）贯穿始终，以应用为重点，讲授基本原理和程序设计技术，使读者得到良好而全面的学习。

该书很适合于工程技术人员自学，对高等院校非计算机专业的高年级学生和研究生来说，是一本恰到好处的教材，对广大的计算机工作者来说，是一本有益的参考书。

张钟俊
于上海交通大学

前 言

在校本科生、研究生，从事计算机在各领域应用的科技工作者，常常会陷入不善于设计程序，或设计的程序质量不高的困境。为此，我们把各种程序设计语言、计算机导论、数据结构、操作系统、编译原理、软件工程等课程中程序设计技术的精华，用程序结构（数据结构、控制结构）这条主线串起来编著了本书，送去我们的一片温馨。

60年代后期出现的“软件危机”以及1968年戴杰克斯特拉的GOTO有害论的炮声，曾促使人们向传统程序设计方法开战。过后不久，他的著名著作《结构程序设计》，给出了结构程序的原理和方法。几乎同时，N·沃思的“自顶向下逐步求精”方法，给出了设计大型程序系统的“自顶向下”，“分而治之”的设计过程。大型复杂程序设计中使用的模块化程序设计方法，又成了70年代以来程序设计方法学的重要研究课题。1967年夫劳艾得用断言方法证明程序框图的正确性，从另一方面开辟了一个程序设计研究的新途径，随后的霍尔公理以及戴杰克斯特拉的谓词转换器，形成了一个程序正确性证明的新天地。从数据结构角度看，递归数据结构的实现，抽象数据类型的引入，特别是抽象数据类型，人们把它誉为程序设计方法学发展中的一个新的里程碑。逻辑语言 PROLOG 的出现，使得程序设计过程，变成为一个构造函数的过程，无疑对计算机体系和程序设计产生了重大影响。PROLOG 可能成为第五代机的核心语言，而使用

PROLOG 进行程序设计的研究已成为一个重要的课题。

本书以程序的结构（即控制结构和数据结构）为主线，讲述了结构程序设计方法，递归程序设计方法，程序的证明与测试；阐明了各种数据结构；给出了结构程序设计语言 PASCAL。以期从方法到工具，使读者获得程序设计技术良好而全面的技能。

本书的突出特点是：

1. 突出了结构化程序设计技术：“自顶向下”、“逐步求精”、“结构模块”等方法。使得读者在受到程序设计的方法、技巧和风格等基础训练之后，就可以设计出易阅读、易修改、可靠性高的实用程序。

2. 结构严谨，逻辑性强。本书依程序结构为主线，阐明了程序设计技术的精髓，探讨了程序设计技术的未来。

3. 本书语言通俗，说理透彻，选例得当而有趣，既有广度，又有深度。因此，特别适合教学和自学。

如今，程序设计技术已成为计算机科学的重要组成部分，是软件工程学的核心。“海阔凭鱼跃，天高任鸟飞”，你可以在这一广阔天地里，跃入一个无限美好的信息时代。

参加本书编写工作的有孔俊宝教授，钟竞副教授，梁秉岑副教授。中国人民解放军驻洛阳第二军事代表室高级工程师谢万田仔细审校了全部书稿。

在本书的编写过程中，我们始终得到了中国科学院学部委员、上海交通大学张钟俊教授的全面指导和帮助，他不但校阅了书稿，还亲自为本书作“序”。吉林工业大学的陈永昌教授、成都电子科技大学的龚天富副教授、江苏工学院的查杰民副教授，都为本书的编写给予了有力的支持，

在此，对他们致以诚挚感谢！

鉴于我们的水平，肯定会有疏漏和不周之处，敬请专家和读者批评指正。

编 者

1991年5月

目 录

第一章 绪论.....	(1)
第一节 程序设计的发展.....	(1)
一、软件危机.....	(1)
二、程序设计技术的发展.....	(1)
第二节 程序设计的一般途径.....	(5)
一、程序设计的概念.....	(5)
二、程序设计的一般过程.....	(8)
第三节 结构程序设计的一般概念.....	(8)
一、什么是结构程序设计.....	(8)
二、自顶向下逐步求精.....	(9)
第四节 程序结构.....	(11)
一、程序的数据结构.....	(11)
二、程序的控制结构.....	(12)
习题一.....	(15)
第二章 结构程序设计语言 PASCAL.....	(17)
第一节 PASCAL 语言概述	(17)
第二节 PASCAL 程序结构	(18)
一、引例.....	(18)
二、程序结构.....	(19)
第三节 PASCAL 语言的词汇集.....	(21)
第四节 说明与数据类型.....	(23)
一、说明.....	(23)
二、数据类型	(24)

第五节 语句.....	(36)
一、简单语句.....	(36)
二、结构语句.....	(40)
第六节 子程序.....	(45)
一、过程.....	(46)
二、函数.....	(50)
习题二.....	(52)

第三章 数据结构概览..... (53)

第一节 概述.....	(53)
一、数据结构.....	(53)
二、数据结构的发展与地位.....	(55)
第二节 线性表与链表.....	(56)
一、线性表及其操作.....	(56)
二、线性表的向量存储结构.....	(58)
三、线性表的链式存储.....	(61)
第三节 堆栈与队列.....	(64)
一、堆栈及其存储结构.....	(64)
二、队列.....	(67)
第四节 树.....	(72)
一、树的定义.....	(72)
二、树的基本术语.....	(73)
三、树的基本运算.....	(75)
第五节 二叉树.....	(76)
一、二叉树的定义.....	(76)
二、二叉树的性质.....	(76)
三、树转换成二叉树的方法.....	(79)
第六节 遍历二叉树.....	(80)
一、二叉树的存储结构.....	(80)

二、二叉树的遍历.....	(82)
第七节 图.....	(85)
一、图的基本概念.....	(85)
二、图的表示.....	(87)
三、图的运算.....	(91)
习题三.....	(100)

第四章 结构程序设计技术..... (101)

第一节 程序设计的基本原理.....	(101)
一、抽象.....	(101)
二、子目标.....	(105)
三、模块化.....	(105)
第二节 程序系统的生命周期.....	(106)
一、系统分析阶段.....	(106)
二、系统设计阶段.....	(107)
三、系统编码阶段.....	(108)
四、测试阶段.....	(108)
五、运行和维护阶段.....	(109)
第三节 结构化的控制结构.....	(110)
一、过程.....	(110)
二、函数.....	(112)
三、一些新的控制结构.....	(113)
第四节 自顶向下程序开发技术.....	(114)
一、自顶向下与自底向上.....	(114)
二、程序模块.....	(115)
三、模块结构图.....	(116)
四、模块开发的顺序.....	(119)
第五节 逐步求精法.....	(120)
一、逐步求精算法设计的基本思想.....	(120)

二、精细化技术.....	(121)
三、逐步求精方法设计举例.....	(123)
四、讨论.....	(130)
习题四.....	(131)
第五章 递归技术.....	(132)
第一节 递归的概念.....	(132)
一、概念.....	(132)
二、例举.....	(133)
三、讨论.....	(135)
第二节 递归与迭代.....	(136)
一、递归与迭代的关系.....	(136)
二、递归程序的执行过程.....	(139)
第三节 递归程序设计.....	(142)
一、递归程序的模型.....	(143)
二、例举.....	(144)
第四节 递归算法的模拟.....	(156)
一、采用叠代算法.....	(157)
二、“末尾递归”的消除.....	(158)
三、利用堆栈消除递归.....	(159)
第五节 递归化.....	(164)
一、递推的递归化.....	(164)
二、数据结构的递归化.....	(169)
习题五.....	(173)
第六章 程序的正确性.....	(175)
第一节 程序的测试.....	(175)
一、程序测试的一般概念.....	(175)
二、白盒测试法.....	(178)

三、黑盒测试法.....	(181)
第二节 夫劳艾得-霍尔规则公理方法	(185)
一、基本概念.....	(185)
二、夫劳艾得不变式断言法.....	(188)
三、霍尔公理化方法.....	(191)
第三节 戴杰克斯特拉最弱前置条件方法.....	(195)
一、谓词转换器的基本概念.....	(195)
二、关于程序证明的一个演绎系统.....	(198)
三、程序的形式推导开发技术.....	(204)
习题六.....	(207)
参考文献.....	(209)

第一章 绪 论

第一节 程序设计的发展

一、软件危机

电子计算机自1946年问世以来，发展十分迅速，在短短的40多年中，已经历了从电子管、晶体管、集成电路、大规模集成电路四个时代，向超大规模集成电路挺进着。这个本世纪科学技术最卓越的成就之一的电子计算机，它的研制水平、生产规模、推广应用的程度，已成为衡量一个国家现代化水平的重要标志。可以说，计算机发展的源泉在于应用，计算机事业的核心还是应用。

60年代中、后期，计算机的应用已相当广泛，而计算机软、硬件的规模也愈来愈大。而那时，程序设计技术还很落后，没有人从宏观上对程序设计方法进行研究。许多人仅满足于写出一个可执行的程序，因此，许多大型软件，常常是质量低劣，可靠性差，价格昂贵，供不应求，浪费了许多人力和物力。此种局面，阻碍着计算机和计算机应用事业的发展，人们把程序设计的这一困境，称之为“软件危机”。

二、程序设计技术的发展

如何走出“软件危机”这条谷底呢？人们从两个方向进行着探索：一是研制各种适用性强的高级程序设计语言；二是探索怎样设计程序，设计程序的基本指导思想是什么，采用什么样的程序设计方法为好等问题。

（一）程序设计语言

1. 机器语言及汇编语言

50年代用0与1代码的机器指令代码,进行手编程序,又称机器语言,这时,程序设计是既繁杂、费时,又易出错的劳动。汇编程序的出现,虽然在助记、直观等方面有了提高,但本质上仍然是面向机器的程序设计语言。

2. 高级语言

最早成功地实现的程序设计语言是FORTRAN语言。1954年,IBM公司的一个研制小组在纽约成立,1957年发表了关于FORTRAN语言的第一篇报告。第一个高级语言FORTRAN的出现,大大简化了程序设计。用高级语言编写程序,是面向问题的。使用高级语言可以避开机器本身的限制,使我们集中精力于算法;同时,高级语言又能自动查出语法错误,因而显示出了强大的生命力。随着计算机应用的普及,高级语言已渗入到各个学科和技术领域,形成了一系列不同的程序风格和不同的服务对象的专用语言和通用语言,推动着程序设计方法的变化和发展。迄今已有上千种高级语言在运行,专用语言有数控语言APT,陈列运算语言APL,代数规范语言OBJ,……;通用语言有:FORTRAN、BASIC、ALGOL、COBOL、LISP、PL/I、PROLOG、PASCAL、C,等语言。

现在许多程序设计语言,引入一些高级措施,以利于程序员在更高的抽象级上进行形式思维和描述。例如,函数型语言PEBBLE和模块化程序设计的MODULA2中,定义了“旌”和“模块”概念的延伸和理论化。应用它们,能更好地发挥程序员的抽象思维,降低程序的复杂性,提高程序的正确性。

(二) 程序设计方法学

1. 关于GOTO语句之争

1968年,北大西洋公约组织于德国召开了一个软件工程师会议,会议分析了计算机软件“危机”的局面,研究了危机的根源,第一次提出用工程化方法解决软件的研制和生产问题,揭开了软件发展史上的新篇章。同年, E.W.戴杰克斯特拉(E.W.Dijkstra)在CACM上,以通讯方式发表了有名的短文“GOTO语句看来是有害的”。这是对传统方法(用程序员的才智去构造程序)的挑战,像捅了马蜂窝一样,引起了轩然大波。他认为: GOTO语句是造成程序一塌糊涂的根源。于是,有人建议取消GOTO语句。其理由是,早在1966年G.杰克皮尼(G.Jacpini)和C.包姆(C.Bohm),已经从理论上证明:任何程序皆可用三种基本构件,通过复合、叠加,象搭积木一样而构成,如图1-1所示。因而,完

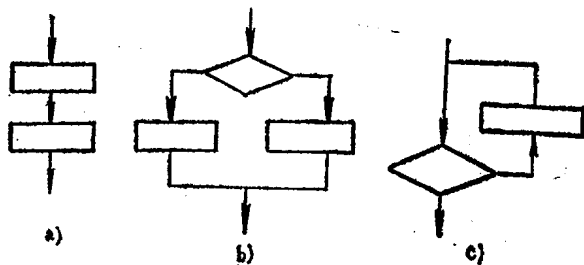


图1-1 三种基本程序构件

a) 顺序 b) 选择 c) 循环

全可以取消给程序带来灾难的GOTO语句。另一些人认为,老语言老软件都使用了GOTO语句,况且完全取消GOTO的程序,效率一般较低,所以主张限制GOTO语句的使用。这场GOTO语句之争,一直到1974年, D.E.克努特(D.E. Knuth)发表了“带GOTO语句的结构程序设计”一文之

后，才被平息。他的基本观点是：不加限制地使用GOTO语句，特别是使用回跳的GOTO语句，会使程序结构难于理解，此时，应限制GOTO语句；另外，为了提高效率，在不破坏程序良好结构的前提下，有控制地使用GOTO语句，还是必要的。

2. 结构程序设计

1969年，戴杰斯特拉在他的著名论文“结构程序设计”中，提出了结构程序设计的原则和方法，希望通过程序静态结构的良好性，来保证程序动态运行的正确性。N. 沃思(N. Wirth)的论文“自顶向下逐步求精”中，提出了“自顶向下”、“分而治之”的大程序设计的原则。其主要思想是：从欲求解的原问题出发，运用科学抽象办法，把原问题分解为若干个相对独立的小问题，依次细化，直至所有小问题皆能解决为止。

这些基本原则和方法，对70年代程序设计方法的发展产生了深刻而积极的影响，成为当时的主导方向和主流。

3. 程序的正确性证明

这是与“结构程序设计”并行发展起来的另一个方向，两者有着紧密的联系。

1967年，R. W. 夫劳艾得(R. W. Floyd)提出用断言方法证明程序框图的正确性。1969年，C. A. R. 霍尔(C. A. R. Hoare)在夫劳艾得的基础上，定义了一个小语言和一个逻辑系统，此系统有程序公理和推导规则，目的在于证明程序的部分正确性。1976年，戴杰斯特拉提出了最弱前置条件和谓词转换器的概念，将部分正确性证明和终止性证明于一体。首次把程序设计从经验、技术升华为科学。

这里，顺便提几点：一是形式化软件开发方法、大型软

件的“模块”程序设计方法，是70年代程序设计方法学的一个十分重要的研究课题；二是抽象数据类型的引入，使程序设计方法学发展到一个新的里程碑。而逻辑程序设计和函数程序设计代表着一个新的研究动向，在下一代计算机中，这类程序设计方法可能占据突出的位置。

总之，程序设计方法学，是研究和创造程序的学问，是研究关于问题分析、环境模拟、概念获取、需求定义以及这种描述变换、细化和编码成为机器可接受的一般表示方法。

第二节 程序设计的一般途径

一、程序设计的概念

(一) 什么叫程序设计？

1. 指令 指令是计算机的 CPU 所能执行的操作命令。

2. 程序 程序是有序的计算机指令的集合，计算机一步一步地执行这个指令序列，就能完成期望它干的事情。

3. 程序设计 编制程序使计算机能正确运行并完成要求它干的全部工作叫程序设计。或者说，排定以时间为进程必须完成的各种事情叫程序设计。

(二) 程序设计的特点

1. 构造性

程序设计如同工程师绘制蓝图、编制工艺规程，要先“构造程序”。根据求解的问题，构造出一个程序实体，然后上机运行。构造性使我们编程序时比较自由，所以，程序设计的结果存在多样性。构造性决定了程序设计的工程性质，因而也决定了程序难于建立统一的、定量的质量标准。