

结构化程序设计方法 与 PASCAL 语言

刘金宏 赵宏 张伟 编著

中国铁道出版社

1



PDG

内 容 简 介

本书系作者在北方交大长期从事结构化程序设计和 PASCAL 语言教学讲稿经过加工整理而成。全书内容明显贯穿着三条主线：结构化程序设计方法主线、数据结构主线和算法设计主线。PASCAL 语言的程序结构、数据类型和语句功能全面体现结构化程序设计能力的这三条主线之中。

第一章介绍软件开发的过程、方法和工具；第二章到第八章介绍 PASCAL 语言程序设计；第九章为常用数值算法的 PASCAL 实现和简单误差理论；第十章为常用非数值算法的 PASCAL 实现。

本书可作为大专院校计算机应用专业和非计算机专业的教材，也可供各类培训班、中专学生和自学者选用。



前 言

由 E. W. Dijkstra 和 C. A. R. Hoare 首先定义的结构化程序设计概念是由 N. Wirth 教授设计的 PASCAL 语言首先实现的。结构化程序设计方法仍是目前软件开发中普遍采用的分析设计方法。因此我们可以毫不夸张地说 PASCAL 语言是程序设计语言发展中的一个里程碑。

PASCAL 语言丰富的数据类型、严谨的语法结构以及 PASCAL 程序高清晰的自我说明性也是其它程序设计语言不全具备的。尽管其它程序设计语言也正在向结构化方面发展,而通过学习 PASCAL 语言来学习结构化程序设计方法才是原汁原味。

正是由于 PASCAL 语言的优秀品质,至使像操作系统原理、数据结构、算法设计与分析等重要计算机课程中的算法描述都采用 PASCAL 语言或类 PASCAL 语言。正在发展起来的第四代语言不少也爱以 PASCAL 语言为背景。显而易见,步入计算机世界的第一步就抓住“结构化程序设计方法”并掌握 PASCAL 语言是一本万利之举。

在长期的教学实践中我们接触过不少有关 PASCAL 语言的教材。从总体上讲我们认为存在着一死、二细、三缺少的问题。一死指有些教材是用例题对语言文本文法进行解释;二细指有些教材过细过偏,有向科研专著发展之趋势;三缺少指缺少内容与方法的融合、缺少内容与教法和学法相融合、缺少内容的编排与认知规律的相适应。“结构化程序设计方法与 PASCAL 语言”一书的目的是想在力避这“一死、二细、三缺少”方面作一点探索性工作。为此本书突出以下特点:

第一,体现学为主体教为主导的教学思想。本书从内容编排上力避满堂灌、适宜教师启发引导教学,在课堂上便于把学生推到主动

参与的地位。

第二,体现具体——抽象——具体的认知规律。本书各章先以具体实例的 PASCAL 语言解法开始,抽象出 PASCAL 语言相关语法概念,再把学生推到举一反三地解决具体程序设计问题的环境,加之深入浅出的叙述引导,便于教也便于学。

第三,体现编程能力四要素。根据算法+数据结构=程序的科学公式,本书始终强调结构化的分析设计方法、抓住算法和数据结构对程序的作用、努力创造上机操作环境,从四方面反复强化编程能力的培养。

此外,“结构化程序设计方法与 PASCAL 语言”一书还体现了对学生的低起点、高目标的要求。本书要求读者仅具备初等数学知识,而对计算机基础知识可以零为起点。本书从内容编排的循序渐进性、主线清晰性、重点内容突出全面保证了对读者的低起点高目标的要求。

本书前八章为结构化程序设计方法和基本 PASCAL 语言,最能体现本书特色。可作为课堂教学的主体。第九章是常用数值算法和误差理论初步。供在数值计算方面强化学生能力选用。第十章是非数值算法——数据结构处理。供在非数值算法方面强化学生能力选用。把数值算法和数据结构部分从基本 PASCAL 中分立出来是对这个问题进行“结构化分析”的结果。它适应了本书对读者低起点高目标的要求,给教师和学生教和学的余地,便于发挥他们的主观能动作用。

本书前八章由刘金宏同志执笔并由他统编了全书。第九章由赵宏同志执笔。第十章由张伟同志执笔。我们的出发点是明确的,也尽了最大努力,但毕竟受水平所限,疏漏错误在所难免,恳请读者不吝赐教。

编著者

1996.2

目 录

第一章 程序设计原理、工具及运行环境	1
1.1 简单问题程序设计	1
1.2 较复杂问题程序设计	8
1.3 大型软件系统开发——软件工程原理	13
1.4 结构化程序设计原理	25
1.5 PASCAL 语言特点及其程序结构	29
 第二章 PASCAL 顺序结构程序设计	 42
2.1 引 例	42
2.2 实型、字符型和标识符常量	49
2.3 调用 Write(及 Writeln)过程制表初步	55
2.4 PASCAL 标准函数库	57
习 题	61
 第三章 PASCAL 选择结构程序设计	 63
3.1 引 例	63
3.2 布尔数据类型	67
3.3 选择结构程序设计	70
3.4 运输费用的计算——CASE 语句	76
3.5 枚举数据类型	82
3.6 子界数据类型	88
3.7 类型间相容	90
习 题	92

第四章 PASCAL 循环结构程序设计	94
4.1 引 例.....	94
4.2 WHILE . . . DO 和 REPEAT . . . UNTIL 循环结构	100
4.3 FOR . . . TO(或 DOWNT0) . . . DO 循环结构	106
4.4 构造数据类型 I —— 数组	117
4.5 数组类型应用举例	127
4.6 构造数据类型 II —— 集合	132
4.7 一个非结构化语句——GOTO 语句	142
习 题.....	145
 第五章 构造数据类型 II —— 记录和文件.....	146
5.1 引 例	146
5.2 记录类型应用举例	153
5.3 FILE 类型文件	157
5.4 Text 类型文件	167
习 题.....	174
 第六章 PASCAL 程序的模块设计——过程和函数	175
6.1 PASCAL 语言的过程和函数	176
6.2 再论过程和函数的参数	188
6.3 设计结构化的 PASCAL 程序系统	194
习 题.....	198
 第七章 内存储器的节省与动态数据结构.....	201
7.1 压缩字符型数组	201
7.2 变体记录	203
7.3 动态数据结构	208
7.4 单链表及其检索、插入、删除操作	216

7.5	单链表操作程序系统	222
7.6	单链表应用举例	224
7.7	循环链表和双向链表初步	229
习 题		231
第八章	递归过程和递归函数初步	233
8.1	递归算法与递归定义	233
8.2	Hanoi 塔问题的递归解法	235
8.3	一般问题的递归解法	237
8.4	间接递归及其调用	238
习 题		240
第九章	常用数值算法与误差理论初步	241
9.1	算法与误差	241
9.2	非线性方程求解	245
9.3	数值积分	255
9.4	其它算法	263
第十章	数据结构处理	268
10.1	线性表问题处理	268
10.2	非线性问题处理	284
10.3	查找及排序操作	294
习 题		299
主要参考文献		300

第一章 程序设计原理、工具及运行环境

本章涉及到计算机科学领域里相当广泛的内容。用短短一章的篇幅要详述这些领域的知识和技能具有相当的难度,因而本章必然浓度极高。对于一个计算机用户来说,他无时不在某种计算机系统环境中运用着本章所要介绍的原理和工具以设计出高质量的软件产品,因此本章对于计算机用户来说又是至关重要的。要求初学者在学习本章时掌握主要概念和思路,而在后续章节的学习中一定要反复加深对这些概念的理解,在反复运用这些原理和工具的实践中达到自如,而切忌死记硬背一些条条框框。

1.1 简单问题程序设计

1.1.1 程序设计与程序设计语言

程序(PROGRAM)是日常生活中司空见惯的概念。一个会议的议程,一场演出的节目单,一次宗教仪式,接发一趟列车的操作过程,解一个二次方程的步骤都是常见的程序。这些具体的程序有如下共性:它们都由人的一个一个活动组成,有序地完成每一个活动就实现了整个程序的目标。因此 PROGRAM 这个概念可以粗略地认为是以时间为进程为达到某一目标的一步一步人类活动的集合。

PROGRAM 是程序设计的结果。在执行程序前必先排定程序。排定以时间为进程必须完成的各步活动就叫程序设计(programming)。

人执行一个程序可以灵活机动。例如一个节目可以因其演员迟到或所需设备临时故障而暂和后面的节目对调;一个排好的科研计划可以根据随机因素的变化进行人员和进度调整;一个排好的程序在执行过程中发现错误可以改正,暴露出不足可以弥补。这

一切都是因为人具有智能性决定的。

计算机只能按人给它的程序行事。一个错误程序让计算机执行一百遍和执行第一遍没什么区别。错误永远属于设计程序的人，计算机只会听令行事，因此确保提交给计算机执行的程序是正确有效的是程序设计人员的最重要任务。

用计算机懂得的语言编程序才能达到令计算机替人算题的目的。这个道理十分简单，否则岂不是对牛弹琴？计算机语言是程序设计最重要的工具。自计算机诞生以来，计算机语言已经发展到了第四代。

第一代计算机语言——机器语言。因为计算机只认识 0,1 两个字符，因此机器语言编的程序全部是由 0,1 表达出来。可想而知人用这种语言编程序是多么的繁琐，人读这种程序又是多么困难。此外这种语言还是严格依赖机种的，编出的程序没有通用性。目前已没有人直接用这种语言编程序了。但机器语言必竟是机器能执行的唯一语言，因此后代各种语言编的源程序(Source)最终都要通过语言处理程序加工成等价的机器语言程序——目标程序(Object)后才能由计算机来执行。我们将学的 PASCAL 语言也是如此。

第二代计算机语言——汇编语言。汇编语言是对机器语言进行符号化的结果。符号化就是不用 0,1 而使用英文文字的缩写表达语言。由于向自然语言迈进了第一步而稍微增加了点可读性，编程序也稍微方便了一点。然而汇编语言对机种的依赖性旧如机器语言，汇编语言程序没有通用性。然而汇编语言由于离机器语言较近，因此其运行效率较高，对硬件操作较灵活。目前编制系统软件、自动控制方面软件、某些接口部件软件以及特别注重运行效率的软件仍采用汇编语言。汇编语言源程序必须经过汇编程序加工成等价的目标程序才能执行。

第三代计算机语言——高级语言。汇编语言再向自然语言前进则发展到了高级语言阶段。高级语言是脱离机种的，只要配置适当的语言处理程序——编译程序，就能将用高级语言编的源程序编译成等价的目标程序，高级语言程序具有通用性。另外高级语言

是面向过程的语言(Procedure-Oriented)。这意味着使用高级语言编程程序的本质工作是用计算机语言的语句描述解题过程的每一步。PASCAL 是高级语言。FORTRAN、BASIC、COBOL、C 等等大多数流行语言都属第三代语言。

第四代计算机语言——非过程化语言。非过程化语言的设计目标是使用户摆脱对解题过程的繁琐描述,而仅仅向计算机提出要解决的问题是什么——面向问题的语言。FORH 是典型的非过程化语言。真正的并行运算语言 ADA 也属于非过程化语言范围。目前非过程化语言正面临着大发展的前夕。此外在计算机界还有配合第五代计算机的第五代语言,甚至更高代语言之说,我们认为这多带有商业广告性。然而对于一个计算机工作者密切注视计算机语言的发展动向是完全应该的,不管第几代语言编的源程序最终都要加工成目标程序才能由计算机执行是永恒的。

请搞清下列概念:程序,程序设计,计算机源程序,目标程序,汇编语言程序,汇编程序,编译程序,面向过程的语言,非过程化语言。

1. 1. 2 用 PASCAL 语言编程序

例题:编 PASCAL 程序,用公式 $y=x^2-2x+3$ 计算 $x=0,1,2,\dots,10$ 所对应的 y 值。

对于这样一个能用数学公式准确表达数学模型的小题目一般经过下面三步就可编出正确有效的 PASCAL 程序:

- 选定算法并用适当工具描述该算法;
- 编码——用 PASCAL 语句描述选定算法;
- 上机调试或测试程序。

我们选定一种伪代码(Pseudo Code)来描述选定的算法:

1. 置 x 为 0(下界);
2. 置 x 的上界 n 为 10;
3. 当 $x \leq n$ 时,做
 - (a)用公式计算 y 值;
 - (b)输出一组 x 和 y 值;

(c)使 x 值增加 1;

4. 停止。

这里有必要对一种程序设计和描述工具——伪代码进行简要说明。伪代码是以约定的文字表现程序结构。一般均为程序设计语言和自然语言的混杂体,使用方便但没有标准化格式,伪代码程序最终都要转化成某种高级语言。有关伪代码的详细知识请参阅程序开发技术或软件工程方面的书籍。本书不打算以伪代码作为分析设计程序的主要工具。

将伪代码程序转化成 PASCAL 语言程序 1—1:

```
1  PROGRAM example1;  
2  VAR x,y,n: Integer;  
3  BEGIN  
4      x:=0;  
5      n:=10;  
6      WHILE x<=n DO  
7          BEGIN  
8              y:=x*x-2*x+3;  
9              writeln(x,y);  
10             x:=x+1  
11         END  
12 END.
```

程序 1—1

程序每行开头的行号不是语句的组成部分。加入行号纯为叙述方便,以后的程序均如此。PASCAL 程序的第 4,5,6,8,9,10 行分别是伪代码程序 1、2、3、3.1、3.2、3.3 转化的结果。PASCAL 程序的第 1 行称程序首部,第 2 行构成程序说明部分,第 3 和 12 行称整个程序可执行部分并包含有停机功能,第 7 和 11 行将作为一个顺序块执行的 8,9,10 三行当成一个复合语句作为 DO 的域反复执行,这些都是 PASCAL 程序结构要求必须遵循的规定,在此不求甚解,要在遵守中自成习惯。

下一步就是将这个 PASCAL 程序录入计算机,经编译、连接、

装入并运行之。在录入、编译、连接以至运行的每一步都可能发现错误,这时需要调试修改错误再从相应的地方重新开始上述过程,直到完全无错拿到正确结果为止。

思 考 题

1. 编一个公式明确的题目的程序分几步?
2. 一 PASCAL 程序可认为由哪三部分构成?
3. 谈一谈你上机运行程序 1—1 的过程和体会。

1.1.3 PASCAL 程序上机环境

硬件和软件两部分构成的现代计算机系统为用户提供基本应用环境。硬件可以被看成是计算机系统的躯体,软件就是其灵魂。没有硬件便执行不了基本功能,软件也无处发挥其作用;而没有软件硬件只不过像一具僵尸什么事也干不成。

1. 硬件环境

(1) 从数据处理过程看计算机硬件组成(图 1—1)

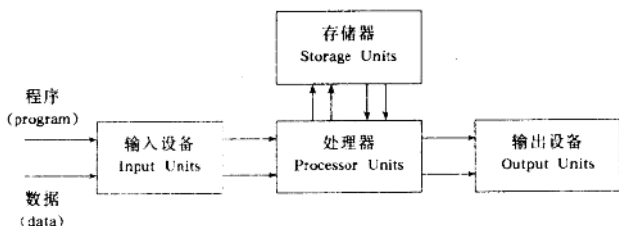


图 1—1 计算机系统硬件组成示意图

• 输入设备: 程序和数据是通过输入设备送入处理器的。最常用的输入设备就是键盘, 当然根据计算机的不同用处可以有各种各样的输入设备。

• 输出设备: 程序运算的结果通过输出设备从处理器里输出。根据不同的用处可以连接各种输出设备, 而常用的输出设备是显示器、打印机和绘图仪等。

• 存储器: 这里指的是外存储器或称辅助存储器。有多种外存

储器,但最常用的是磁盘存储器,常分为硬盘和软盘两种。硬盘容量可从几十兆字节到几百兆以至几千兆字节不等,其容量大、速度较软盘快但价格较贵;软盘分 5 英寸和 3 英寸两种,5 英寸容量 1.2 兆字节,3 英寸容量 1.4 兆字节。此处 1 字节按 8 位二进制位计算,一个二进制字节可存放一个二进制 0 或 1。

- 处理器:处理器由内存和中央处理器(CPU)构成,它也称主机(对大、中、小型机而言),是计算机硬件的核心。程序和数据流动必经处理器。

(2)处理器结构(图 1—2)

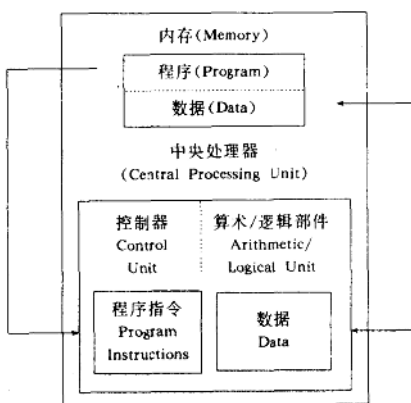


图1—2 处理器结构示意图

- 内存(Memory):是存储运行中的程序和数据部件。程序指令要送到控制器去分析处理,数据要送到运算器参加运算并将运算结果(或中间结果)送回内存。

- 控制器(Control Unit):从内存取程序指令进行分析并对其相关部件发出操作控制信息。

- 运算器(Arithmetic/Logic Unit):按控制器的控制信息取数、进行算术或逻辑运算、将运算结果送到内存。

控制器和运算器合称中央处理器 (Central Processing Unit——CPU)。当给出运行命令以后因程序是事先存储在内存中的,因此执行过程是自动进行的。

2. 软件环境

用户用 PASCAL 语言源程序上机时应设置一台 PASCAL 语言计算机系统,如图 1—3 所示。

图 1—3 表示用户要在操作系统和 PASCAL 编译程序两个系统软件环境支持下才能运行 PASCAL 语言源程序。

(1) 操作系统 (Operating System)

操作系统是管理和协调计算机系统中全部硬、软件资源的一个管理软件。对用户来说它是使用计算机系统资源的唯一窗口。用户要向计算机申请每一种资源的请求命令都必须发送给操作系统并由操作系统分配协调,用户上机的第一件事就是用操作系统启动计算机后,计算机才处于可用状态,操作系统在开机整个过程中常驻内存。MVS 是典型的运行在大型机上的操作系统,DOS、UNIX、OS/2 等是运行在小型、微型机上的操作系统。

(2) PASCAL 编译程序 (Compiler)

是操作系统管理的软件资源之一。它的功能是将 PASCAL 语言源程序翻译成等价的目标程序以供连接之后执行。

除操作系统和编译程序之外,用户在将 PASCAL 语言源程序录入计算机时还经常用到一个实用程序 (Utility)——文本编辑器 (Editor),它的功能是允许用户录入源程序并修改源程序。

3. PASCAL 源程序上机过程

PASCAL 语言源程序上机一般要先经过录入、编译、连接最后进入装入运行阶段。每一阶段出现错时都要修改源程序,修改后的源程序必须从编译重新开始上述过程,而修改工作要使用编辑程



图 1—3 PASCAL 语言
计算机示意图

序完成。这一过程我们由图 1—4 表示。

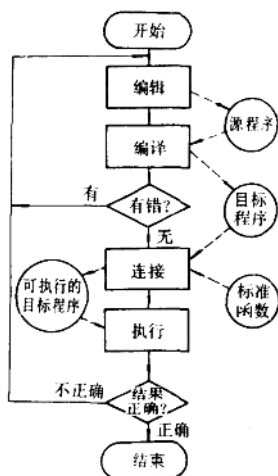


图 1—4 PASCAL 源程序上机过程

思 考 题

1. 现代计算机系统由哪两部分组成?
2. 叙述下面软硬件的功能:输入设备,输出设备,外存储器,内存,控制器,运算器,操作系统,编译程序,编辑程序。
3. PASCAL 源程序上机过程分几个阶段?

1.2 较复杂问题程序设计

这里说的较复杂问题是和类似 $y=x^2-2x+3$ 那样简单的问题相比较,或者题目用较多的数学表达式且每个表达式间关系较复杂,或者问题根本就没有现成的数学式表达,其解法需要通过设计者自行分析出一套表达式来解题,这种较复杂问题的解法一般要遵循四个解题步骤:建立数学模型,选定计算方法,设计和编码以及测

试,显然包含着简单问题的解题步骤,也加了一些新的步骤和内容。

1. 建立数学模型

例题:有 A、B、C、D、E、F 六名学生,有留校、北京、南京、贵州、四川、沈阳六个去向。每个学生可填两志愿。用计算机完成分配问题。

显然此题无现成公式可用,这就必须自行建立数模,办法是让学生填志愿表如下:

A 北京 南京

B 留校 贵州

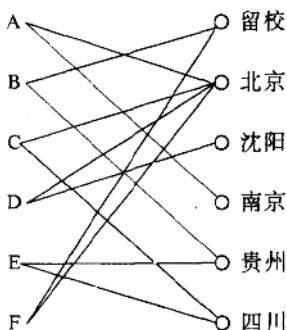
C 北京 四川

D 北京 沈阳

E 贵州 四川

F 留校 北京

根据志愿表构造一张图如下:



这便将分配问题映射成一个解决图问题的数学模型。显然只有懂图论并有相当分析经验的人才能做到这一步。构造数学模型是程序设计中最复杂、最困难的一步。它需要具备强有力的数学知识,丰富的经验,专门的知识,因此建模问题远非一个程序员力所能及的,需各路专家共同完成。一般来讲都是将一个实际问题映射到各

种数学领域的解法,因此数学应是计算机工作者永恒的论题。

2. 选定算法

上例构造出了图论问题模型,具体解法还有多种供选择。为简单起见,我们回到 $y=x^2-2x+3$ 的另一个算法来说明选定算法问题。

- ①置 x 为 0;
- ②计算 y ;
- ③输出 x, y ;
- ⋮
- ⑩置 x 为 10;
- ⑪计算 y ;
- ⑫输出 x, y ;
- ⑬停止。

和上一节选择的算法比你肯定说:这个算法不好,上一个算法好。为什么?你一定会说:这个算法冗长罗嗦。很对。用程序设计的术语说好的算法的标准是:

- (1)清晰易懂,结构化程度高;
- (2)效率高——编出的程序占用内存少,运行效率高。

应该牢记:高质量算法是编出高质量程序的前题。

3. 设计和编码

编码就是将选定的算法从非计算语言的描述形式代换成 PASCAL 语言语句形式。这一步在第一节已经做过了。设计实际上是一种分析设计阶段,这对复杂一些以及更复杂的问题是必须的。我们采用的是自顶向下逐步细化的结构化分析方法,这将在后续内容中逐步深入学习。在此我们仅介绍几种常用的分析设计工具。

(1) 自然语言

这是最早描述算法的方法。自然语言的致命缺陷是它的二义性。例如“王老师让小李把他的书拿来。”这句话中“他的书”就有二义性。人可以根据当时的场景确定是谁的书,是哪本书,而对于计算机来说这都是不明确的。计算机语言语句要求无二义性且上下