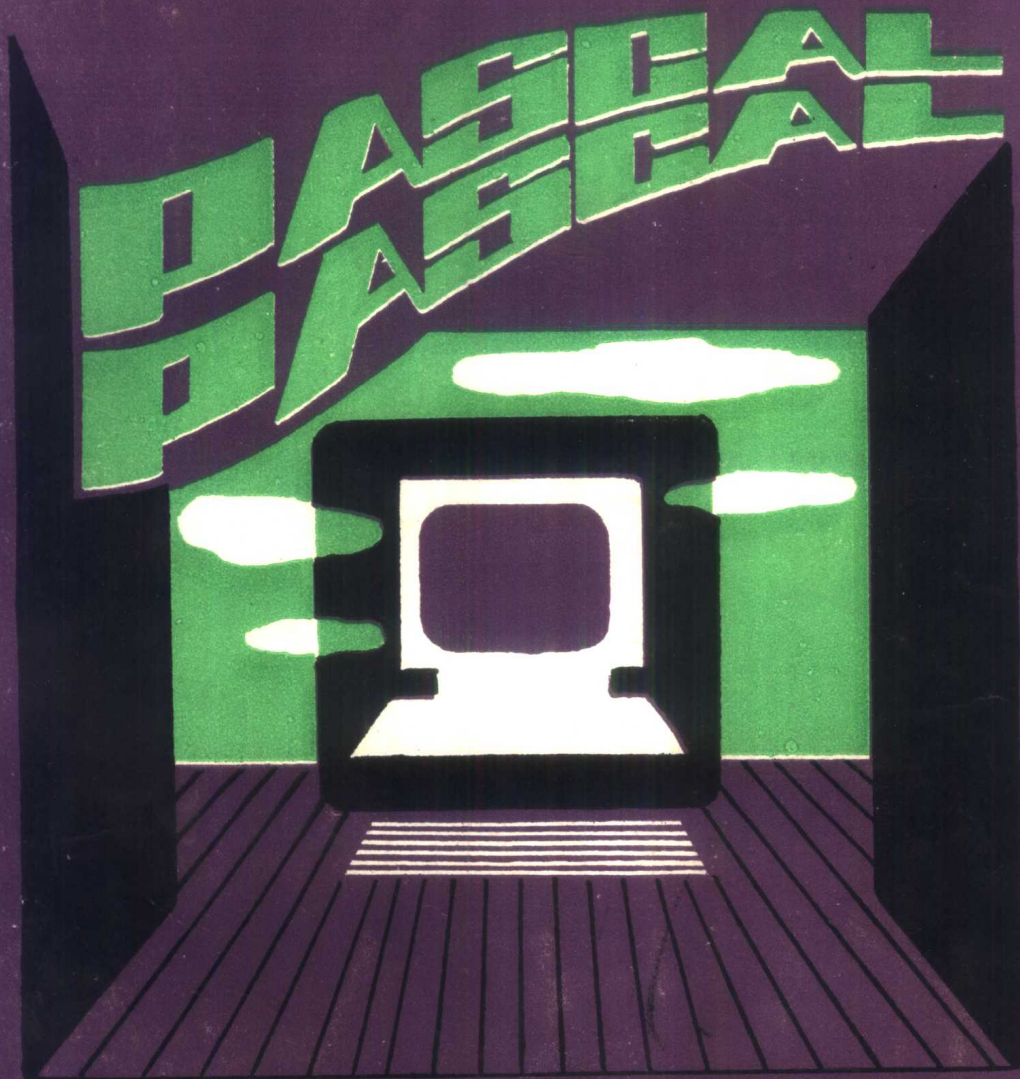


计算机大专系列教材



袁蒲佳 潘国楠 金灿明

PASCAL语言 程序设计

华中理工大学出版社

73.87429
C 297
73.87429
C297

计算机大专系列教材



PASCAL 语言程序设计

袁蒲佳 潘国楠 金灿明

华中理工大学出版社

内 容 简 介

本书是计算机大专系列教材之一。全书共分十章，对 PASCAL 语言的基本概念、数据类型及各类语句的语法和语义做了比较全面的、系统的介绍。根据学以致用原则，对 PASCAL 程序设计方法、风格和应用尤为注重。内容深入浅出，文字力求通俗易懂，便于自学。书末附有学习指导与实验，内容包括课程大纲、学习辅导、习题解答及上机实验。

本书可供各类高等专科学校计算机有关专业作为教材，亦可供高校本科有关专业师生、工程技术人员和其它读者学习、参考。

PASCAL语言程序设计

袁蒲佳 潘国楠 金灿明

责任编辑 唐元瑜

华中理工大学出版社出版发行

(武昌喻家山)

新华书店湖北发行所经销

石首市第二印刷厂印刷

*

开本：787×1092 1/16 印张：23 字数：487 000

1990年12月第1版 1993年7月第4次印刷

印数：11 001—16 000

ISBN 7-5609-0478-5/TP·43

定价：9.90元

(鄂)新登字第10号

《计算机大专系列教材》序

计算机是20世纪最伟大的发明之一。在迎接世界新技术革命的挑战和加快我国四个现代化建设的进程中，大力推广应用计算机具有很现实的和深远的意义。目前，计算机特别是微型计算机的应用已广泛渗透到我国社会和国民经济的各个领域。要使我国计算机的开发和应用进一步向深度和广度发展，以便为我国的社会主义建设带来更大的经济效益和社会效益，其中一个很重要、很迫切的任务就是，要迅速培养一大批主要从事计算机应用工作的专科层次的计算机专业人才。为此，我校已设置了“计算机及应用”专业（专科）。

为满足计算机专科教学工作的需要，我们特邀请和组织华中理工大学计算机系具有丰富教学经验的有关课程的一批骨干教师编写了这套计算机大专系列教材。该系列教材包括《逻辑设计》、《PASCAL语言程序设计》、《IBM-PC宏汇编语言程序设计》、《计算机组成原理》、《数据结构》、《计算机操作系统》、《微型计算机及其应用》和《数据库系统概论》等8种，并均由华中理工大学出版社出版。

该系列教材是参照计算机专业（专科）教学大纲，按照面向应用、重视实践、便于自学的原则和以广泛使用的IBM-PC微机作为教学用机进行编写的。为了使该系列教材成为符合上述原则的统一整体，在教材的编写过程中，对于各书的内容、取材、界面、衔接及格式等问题，都反复地进行了研讨和协商。同时，为了便于读者自学，各书中均编写了“学习指导”方面的内容。因此，该系列教材不仅可供普通高校（专科）计算机类有关专业使用，而且也可供成人高等教育自学考试、函授大学、夜大学、广播电视大学、职工大学等计算机类有关专业使用。同时，该系列教材也是广大工程技术人员和其他在职人员系统地自学计算机知识与技术颇为适宜的读物。

华中理工大学成人教育学院胡润清副研究员、郑青林助理研究员等领导同志具体担负了该系列教材的规划、组织工作。

该系列教材在编、审和出版过程中，华中理工大学计算机系李崇阁副研究员、冯著明副教授等领导同志以及有关课程的教师做了大量的工作，华中理工大学出版社给予了大力的支持与帮助，在此一并表示感谢。

限于水平与经验，该系列教材肯定会有许多缺点和不足，诚望有关专家和广大读者积极提出批评与建议，共同为不断提高该系列教材的质量而努力。

华中理工大学成人教育学院

1990年4月

05128/98

前 言

PASCAL语言程序设计是计算机类专业（大专）的重要专业基础课，本书是参照计算机专业（专科）教学大纲编写的。

PASCAL语言是一种通用的高级程序设计语言。PASCAL语言以其基本结构少但框架优美、表达功能强，数据类型丰富、便于描述复杂的数据对象，编写的程序易读、易维护、易移植、效率高等许多特色而著称，因而自70年代初问世以来被广泛应用于科学计算、数据处理和系统软件设计。目前大多数高等院校计算机类专业，选用PASCAL语言作为讲述结构程序设计、培养学生良好程序设计风格的教材。本书系统地介绍了PASCAL语言的基本概念、数据类型及各类语句的语法和语义。为了帮助读者掌握PASCAL程序设计的基本方法和基本技巧，书中结合应用安排了较多的例题，希望读者用心地阅读这些例题，从中学习一些基本规律，并通过每章后的习题练习和上机实习，受到一定的实际训练。

本书作为计算机大专系列教材之一，在内容的选取、概念的引入、文字的叙述以及例题和习题的选择等方面，都力求遵循面向应用、重视实践、便于自学的原则。全书共分十章，并附有“学习指导与实验”。第一章介绍计算机和程序设计的若干预备知识。第二章介绍PASCAL的基本数据类型。第三~五章围绕PASCAL的三种计算结构——顺序结构、分支结构和循环结构讲述程序设计的基本方法、技巧和风格。第六章集中介绍枚举、子界、数组、记录 and 集合等五种用户自定义数据类型。第七章围过程程和函数介绍子程序设计方法。第八、九两章介绍文件和指针类型及其应用。第十章为了便于学生在不同的环境中上机实习，简要介绍了IBM PASCAL, TURBO PASCAL 以及APPLE PASCAL 的若干特色。学习指导与实验方面的内容包括课程大纲、学习辅导、习题解答与上机操作等。编写这些内容的目的是为了便于读者自学和加强实践环节。

本书由袁蒲佳副教授主编。其中，第一~六、十章由袁蒲佳编写，第七、八、九章由潘国楠编写，学习指导与实验由袁蒲佳和金灿明编写，袁蒲佳统编全稿。在本书的编写过程中，华中理工大学成人教育学院、计算机系以及软件教研室的有关领导和老师们给予了大力支持与帮助，在此一并表示衷心感谢。

由于编者水平有限，书中不妥或错误之处在所难免，殷切希望广大读者批评指正。

编 者

1990年6月于华中理工大学（武汉）

目 录

第一章 计算机与程序设计概述	(1)
1.1 计算机的基本组成	(1)
1.1.1 存储器	(1)
1.1.2 运算器	(2)
1.1.3 控制器	(2)
1.1.4 输入输出设备	(2)
1.2 计算机与二进制	(2)
1.3 程序和程序设计语言	(6)
1.4 PASCAL程序设计语言概述	(7)
1.4.1 PASCAL程序设计语言的特点	(7)
1.4.2 PASCAL程序的结构	(8)
1.5 程序设计语言的语法描述	(10)
习 题 一	(11)
第二章 数据及其类型	(13)
2.1 数据和数据类型概念	(13)
2.1.1 常量和变量	(13)
2.1.2 数据类型	(14)
2.2 常量定义与变量说明	(15)
2.2.1 常量定义	(15)
2.2.2 变量说明	(16)
2.3 标准基本类型	(16)
2.3.1 整数类型	(17)
2.3.2 实数类型	(17)
2.3.3 布尔类型	(19)
2.3.4 字符类型	(20)
2.4 标准函数	(21)
2.5 表达式	(24)
习 题 二	(27)
第三章 简单程序设计	(30)
3.1 赋值语句	(30)
3.2 简单输入输出语句	(31)
3.2.1 输入语句	(32)
3.2.2 输出语句	(34)
3.3 简单程序实例	(35)
习 题 三	(38)

第四章 分支程序设计	(40)
4.1 问题的提出	(40)
4.2 程序流程图	(41)
4.3 IF 语句	(42)
4.4 IF 嵌套和复合语句	(45)
4.5 标号说明与GOTO语句	(49)
4.6 CASE 语句	(52)
习 题 四	(55)
第五章 循环程序设计	(56)
5.1 循环程序的概念	(56)
5.2 FOR 语句	(58)
5.3 WHILE 语句与REPEAT 语句	(60)
5.3.1 WHILE 语句	(60)
5.3.2 REPEAT 语句	(62)
5.4 多重循环	(64)
5.5 综合应用举例	(69)
5.5.1 数制转换问题	(69)
5.5.2 菜单驱动程序	(70)
5.5.3 二分法求根	(73)
习 题 五	(75)
第六章 用户自定义数据类型	(78)
6.1 枚举类型	(78)
6.2 子界类型	(83)
6.3 数组类型	(86)
6.3.1 数组概念	(86)
6.3.2 数组说明	(87)
6.3.3 数组分量访问	(88)
6.3.4 多维数组	(95)
6.3.5 紧缩数组	(101)
6.3.6 字符串	(104)
6.4 记录类型	(107)
6.4.1 记录与记录说明	(107)
6.4.2 记录分量访问	(108)
6.4.3 WITH 语句	(110)
6.4.4 记录数组	(111)
6.4.5 记录的变体部分	(118)
6.5 集合类型	(122)
6.5.1 集合与集合说明	(122)
6.5.2 集合运算	(124)
6.5.3 集合应用举例	(126)
习 题 六	(127)

第七章 过程与函数	(130)
7.1 过 程	(130)
7.1.1 过程说明	(130)
7.1.2 过程调用	(132)
7.2 函 数	(136)
7.2.1 函数说明	(136)
7.2.2 函数调用	(137)
7.3 标识符的作用域	(138)
7.3.1 全程量与局部量	(138)
7.3.2 信息传递	(141)
7.4 参数传递	(142)
7.4.1 值参与变参	(142)
7.4.2 参数的传递	(143)
7.4.3 参数的选择	(145)
7.5 递归与向前引用	(146)
7.5.1 递 归	(146)
7.5.2 向前引用	(150)
7.5.3 过程(函数)相互调用的规则	(151)
7.6 应用实例	(154)
习 题 七	(163)
第八章 文件类型	(166)
8.1 顺序文件	(166)
8.1.1 文件说明	(167)
8.1.2 文件缓冲区变量	(167)
8.1.3 文件与数组的比较	(168)
8.2 文件的基本操作	(168)
8.2.1 文件操作的标准过程和标准函数	(168)
8.2.2 建立文件	(171)
8.2.3 从文件读数据	(172)
8.2.4 文件复制	(173)
8.2.5 文件修改	(173)
8.2.6 文件作为参数传递	(177)
8.3 文本文件	(178)
8.3.1 文本文件的特点	(178)
8.3.2 访问文本文件	(179)
8.3.3 标准文件INPUT和OUTPUT	(184)
8.3.4 输入和输出	(185)
8.4 文件应用实例	(189)
习 题 八	(194)
第九章 指针与动态数据	(196)
9.1 指针类型与指针变量	(197)
9.2 标准过程NEW	(200)

9.2.1 NEW过程	(200)
9.2.2 动态数据结构	(201)
9.3 指针的应用	(205)
9.3.1 队列的插入和删除操作	(205)
9.3.2 栈的操作	(207)
9.3.3 一般链结构的操作	(208)
9.3.4 树	(212)
习 题 九	(216)
第十章 标准PASCAL的扩充	(217)
10.1 APPLE PASCAL和标准PASCAL的主要区别	(217)
10.1.1 长整数类型	(217)
10.1.2 字符串类型	(218)
10.1.3 集合类型	(219)
10.1.4 文件类型	(219)
10.1.5 若干扩充的内部过程与函数	(220)
10.1.6 其它区别	(228)
10.2 IBM PASCAL与标准PASCAL的主要区别	(228)
10.2.1 整常数	(228)
10.2.2 结构常数	(229)
10.2.3 若干类型扩充	(230)
10.2.4 运算符	(234)
10.2.5 控制流	(235)
10.2.6 IBM PASCAL文件的特色	(237)
10.3 TURBO PASCAL和标准PASCAL的主要区别	(241)
10.3.1 整常数	(241)
10.3.2 字节类型	(241)
10.3.3 运算符	(242)
10.3.4 字符串类型	(242)
10.3.5 有类型的常量	(243)
10.3.6 文件操作	(245)
10.3.7 其它区别	(246)
附 录	(248)
附录A ASCII 字符代码表	(248)
附录B 保留字和标准标识符	(249)
附录C PASCAL语法图	(250)
参考文献	(254)
学习指导与实验	(255)

第一章 计算机与程序设计概述

电子数字计算机（简称计算机）是一种能够自动地、高速度地进行大量计算工作的电子机器，它的发明和不断发展是20世纪科学技术的卓越成就之一。今日的计算机已不单纯用于科学计算，它已广泛应用于数据处理、情报检索、生产制造、过程控制和数字通信、图象处理等不同领域，有力地推动着生产、科学技术与人类文明的发展。

本章扼要介绍计算机与程序设计的一些基本知识，包括计算机的基本组成，计算机采用的二进数制，程序和程序设计概念，PASCAL语言概貌以及描述语法规则的语法图。通过本章的学习，可掌握一些与程序设计语言有关的背景性材料，为后续章节的学习奠定基础。

1.1 计算机的基本组成

计算机虽然是部非常复杂的机器，但其主要组成结构却和人用算盘答题所用的器具有十分相似之处。

用算盘答题，参加答题的有纸、笔、算盘和人。算盘是运算部件；纸和笔是用来保存和记录原始数据、中间结果的存贮部件；人是控制部件，负责协调各部分有节奏地工作。

计算机和人用算盘答题所用器具相类似，也是由运算器、存贮器和控制器三个主要部件组成的，我们称它们为主机部分。除了主机部分外，还须有计算机和人进行通信的纽带——输入、输出设备。另外，现代计算机还包括有外存贮器。输入、输出设备和外存贮器等是计算机的外围设备。

1.1.1 存贮器

存贮器是存贮代码的部件。计算机根据指令对数据进行快速的运算和处理，指令和数据必须以二进制代码形式存贮于存贮器中。

内存贮器以存贮单元来划分，每个存贮单元对应一个称为单元地址的顺序编号。形象地说，存贮器好比一个特大型旅馆，有成千上万的房间，每个房间有固定的顺序编号。所谓存贮单元相当于旅馆的单个房间，单元地址相当于房间编号，而指令或数据代码相当于房间中来往的旅客。存贮单元的数量称为存贮量。存贮量是衡量计算机性能的重要指标之一，一般有几万字节到几十兆（1兆=100万）字节，每个字节能容纳8个二进位。存贮单元的长度叫字长，各种计算机的字长并不划一。微型计算机的字长多是两个字节（即16位），超级微型机字长为32位。

存贮器具有接受、保存和发送代码的功能，存贮量越大，能容纳的代码量越多。各种类型的信息，如指令、数值、文字、表格、图形等以某种二进制代码存放于存贮单元中后，一般是不会改变的。从任何存贮单元读出代码后，该代码仍旧留在存贮单元中，但当送入新的代码时，就会把存贮单元中原有的内容冲掉而代之以新的代码。我们把存贮器的这种功能形

象地叫记忆功能。

存贮器分主（内）存贮器和辅助（外）存贮器。主存贮器与运算器直接相连，可以发送代码到运算器参与运算，并能从运算器接受运算结果。外存贮器与运算器一般没有直接的联系，但有较大的存贮量，并且可以和内存贮器成批地进行代码交换。对于具有大量数据或指令的计算问题，内存贮容量不敷使用时，或纯粹出于保存的原因，常可利用外存贮器来存放暂时不用的数据或指令。

1.1.2 运算器

运算器是完成各种算术运算、逻辑运算和其它操作的部件。运算过程中，运算器在控制器的控制下，不断从内存贮器取得数据，经过运算后又把结果送回内存贮器保存。这一切是通过执行存放在内存贮器中的指令系列来完成的。

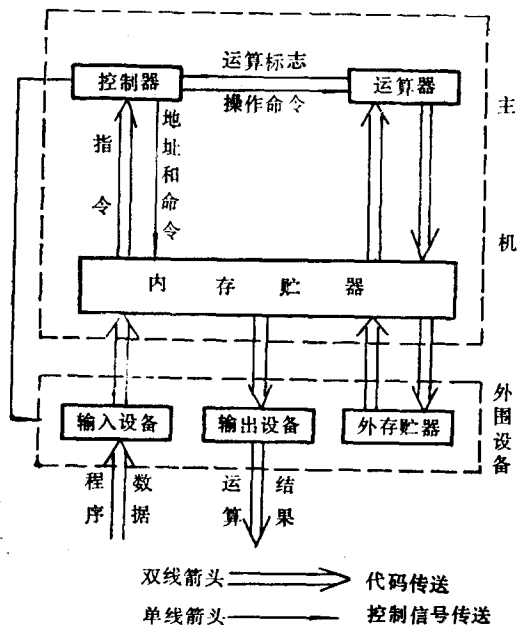


图1.1 计算机组成图

1.1.3 控制器

控制器是控制计算机各个部分协调地进行工作的部件。其主要功能是：

(1) 控制运算器和内存贮器及有关部分自动协调地执行指令系列规定的运算动作，包括从内存贮器取出指令，翻译指令，启动程序（即指令系列）的执行。

(2) 控制计算机外围设备和主机之间进行联系。例如，将程序和数据输入到内存贮器中；将运算结果从内存贮器送至输出设备进行输出；使内、外存贮器成批交换信息。

计算机各组成部分间的联系如图1.1所示。

1.1.4 输入输出设备

输入输出设备是人和计算机进行通信的纽带。输入设备用来输入计算程序和原始数据；输出设备用来输出计算结果。常用的输入设备有光电卡片或纸带输入机、终端键盘、鼠标器等。常用的输出设备有终端屏幕、打印机、卡片或纸带穿孔机等。

1.2 计算机与二进制

计算机与二进制数有不解之缘。要说明什么是二进制数先得从十进制数说起。

日常生活中，普遍采用十进制计数。十进制计数的特点是使用10个数码0~9，采用“逢十进一”进位制。一个十进制整数自右向左的数字依次表示个位、十位、百位、千位……。带小数点的十进制数，小数点左边为数的整数部分，右边为小数部分。小数部分自左向右的数字依次表示十分之一位、百分之一位、千分之一位……。如十进制数

$$\begin{array}{cccccc} 2 & 0 & 4 & 7 & . & 8 & 5 \\ \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\ 10^3 & 10^2 & 10^1 & 10^0 & 10^{-1} & 10^{-2} \end{array}$$

位 位 位 位 位 位

可分解为 $2047.85 = 2 \times 10^3 + 0 \times 10^2 + 4 \times 10^1 + 7 \times 10^0 + 8 \times 10^{-1} + 5 \times 10^{-2}$

十进制是日常生活中广泛使用的,然而不是唯一的一种计数制。人们还经常用到十二进制(如12英寸为1英尺)、十六进制(如老秤16两为1斤)和六十进制(如时间60秒为1分钟,60分钟为1小时)等等。计算机通常采用二进制计数。二进制计数的特点是使用0和1两个数码,采用“逢二进一”进位制。一个二进制整数是一串0和1的数码,自右向左依次表示 2^0 位、 2^1 位、 2^2 位、 2^3 位……。如二进制整数 $1101_{(2)}$ (为了不引起混乱,在数的右下角写上带圆括号的数制号)就是:

$$\begin{array}{cccc} 1 & 1 & 0 & 1 \\ \uparrow & \uparrow & \uparrow & \uparrow \\ 2^3 & 2^2 & 2^1 & 2^0 \\ \text{位} & \text{位} & \text{位} & \text{位} \end{array}$$

这样, $1101_{(2)} = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13_{(10)}$

带小数点的二进制数,其小数点左边为数的整数部分,右边为小数部分。小数部分自左向右的数字依次表示二分之一位、四分之一位、八分之一位……。例如

$$\begin{aligned} &11011.101_{(2)} \\ &= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 27.625_{(10)} \end{aligned}$$

反过来,如何将十进制数转换成二进制数呢?对于十进制整数,可用除二取余法。即将十进制数除以2得到一个商和余数,再将商除以2,又得一新的商和余数,如此继续下去,直到商等于零为止。然后,将所得的各次余数,按先后次序自右向左排列,就是所求二进制的各位数字。

例1: 将 $105_{(10)}$ 表示成二进制数形式。

采用除2取余法有:

$105 \div 2$	商52	余1
$52 \div 2$	商26	余0
$26 \div 2$	商13	余0
$13 \div 2$	商6	余1
$6 \div 2$	商3	余0
$3 \div 2$	商1	余1
$1 \div 2$	商0	余1

于是 $105_{(10)} = 1101001_{(2)}$

十进制纯小数换成二进制小数可用乘2取整法。即每次用2乘纯小数部分,将所得整数(0或1)依次自左向右排列,直到满足所要求的精度或纯小数部分等于零时为止,即得到所求二进制小数部分的各位数字。

例2: 将 $0.6875_{(10)}$ 表示成二进制小数形式。

采用乘2取整法有:

乘2	纯小数部分	整数部分
$0.6875 \times 2 = 1.375$	0.375	1
$0.375 \times 2 = 0.75$	0.75	0
$0.75 \times 2 = 1.5$	0.5	1
$0.5 \times 2 = 1$	0	1

于是 $0.6875_{(10)} = 0.1011_{(2)}$

例3: 将 $0.9_{(10)}$ 表示成二进制小数形式。

采用乘2取整法有:

乘2	纯小数部分	整数部分
$0.9 \times 2 = 1.8$	0.8	1
$0.8 \times 2 = 1.6$	0.6	1
$0.6 \times 2 = 1.2$	0.2	1
$0.2 \times 2 = 0.4$	0.4	0
$0.4 \times 2 = 0.8$	0.8	0
$0.8 \times 2 = 1.6$	0.6	1
$0.6 \times 2 = 1.2$	0.2	1
$\vdots$	$\vdots$	$\vdots$

不难看出, $0.9_{(10)}$ 的二进制表示形式是循环小数。因此, 我们只能按精度要求书写若干位。例如, 若只要求23位, 则是

$$0.9_{(10)} = 0.11100110011001100110011$$

计算机为什么要采用二进制表示法呢? 这是由于二进制有如下优点:

(1) 二进制可以用仅具有两个不同稳定状态的元件来计数码的每一位, 即将一种稳定状态表示为数字0, 另一种表示为1。制造具有两个稳定状态的元件, 比制造多稳定状态的元件容易得多。

(2) 二进制算术运算比较简单。例如

加法 $0+0=0, 1+0=1, 1+1=10$

乘法 $0 \times 0=0, 1 \times 0=0, 1 \times 1=1$

运算很简便, 可以使机器的结构比较简单。

(3) 二进制可以节省存贮设备。例如, 用十进制来表示一个0~999的3位数, 需要用3个元件, 每个元件需要10种状态, 这样, 总共需要30个“元件-状态”。而用二进制表示上面的数, 需要10个元件, 每个元件需要两个状态, 总共只需20个“元件-状态”。10位二进制数可以表示的最大值为1023, 即

$$\begin{aligned} 1111111111_{(2)} &= 2^0 + 2^1 + \dots + 2^9 + 1 \\ &= 2^{10} - 1 \\ &= 1023_{(10)} \end{aligned}$$

二进制数的书写较为冗长, 为了书写的简便, 常使用八进制或十六进制表示。读者不难类推八进制或十六进制表示法。

八进制计数使用八个数码符号0~7, 采用“逢八进一”进位制。如

$$\begin{aligned} 17740_{(8)} &= 1 \times 8^4 + 7 \times 8^3 + 7 \times 8^2 + 4 \times 8^1 \\ &= 8160_{(10)} \end{aligned}$$

十六进制计数使用16个数码符号：0~9和A~F（或a~f），其中A~F（或a~f）依次表示十进制10到15。十六进制数采用“逢十六进一”进位制。如

$$3AF9_{(16)} = 3 \times 16^3 + 10 \times 16^2 + 15 \times 16^1 + 9 \\ = 15097_{(10)}$$

例4：将十进制数782分别表示成二、八、十六进制数形式。

解：按除二取整法可将782表示成二进制形式，即

$$782_{(10)} = 1100001110_{(2)}$$

从二进制转换成八或十六进制是十分容易的，只要3位一并或4位一并即成，即

$$782_{(10)} = 1,100,001,110_{(2)} = 1416_{(8)}$$

$$782_{(10)} = 11,0000,1110_{(2)} = 30E_{(16)}$$

在计算机里，整数和实数具有不同的内部表示，它们分别由计算机硬件所支持。

整数在计算机里的内部表示，包含数符和尾数两部分，即

数符	尾数
----	----

其中数符表示数的正负，占1位，0代表正，1代表负。对于字长为16位的计算机，如APPLE I、IBM-PC等，常用一个存贮字（即两个字节）来表示一个整数。这时，可表示的最大整数是

$$0 \quad \underbrace{111 \ 111 \ 111 \ 111 \ 111}_{15 \text{位尾数}}$$

其十进制数值是 $2^{15} - 1$ ，即32767。整数的取值范围是 $-32767 \sim 32767$ 。

实数的内部表示通常采用浮点表示形式。例如，可将任意实数R表示成下列形式：

$$R = \pm d \times 2^{\pm P}$$

其中：d称为R的尾数，它表示小数部分，d前的符号为数符；P称为R的阶码，P前的符号为阶符。根据这一表示形式，可建立实数的内部表示为：

阶符	阶码	数符	尾数
----	----	----	----

对于字长为16位的计算机，常用两个相邻的存贮字即4个字节来表示1个实数，其中阶符和数符各占1位，阶码和尾数各占一定的位数（比如阶码占7位，尾数占23位），这依赖具体的机器。

例5：十进制数2.75可以表示成

$$2.75 = 2^2 \times 0.1011_{(2)}$$

也可表示成

$$2.75 = 2^3 \times 0.01011_{(2)}$$

或

$$2.75 = 2^4 \times 0.001011_{(2)}$$

等等。

可见，浮点表示并不是唯一的。当小数部分的最高位为1时，称为规格化表示。上面的第一种就是2.75的浮点规格化表示，其内部代码表示象这样：

0	0000010	0	10110000000000000000000
阶符	阶码	数符	尾数

小数点“.”在机器中实际是不表示出来的，形式上定在尾数的最高位之前。然而，由于阶码等于2，所以数的实际小数点在尾数左起第3位之前。

综合上述内容，我们可得出以下三点重要结论：

(1) 整数和实数的机内表示，其代码形式，存储长度及关于值的解释是完全不同的。带小数的数表示成实数（浮点）形式，但整数（既可表示成二进制整数形式，也可表示成实数（浮点）形式。例如100，表示成整数形式时，其代码是

0000000001100100

表示成实数浮点形式时，其代码是

0 0000111 0 1100100 ... 0

└───┘ └───┘

阶码 尾数

两种表示形式截然不同。

(2) 不管是整数还是实数均有一定的值的表示范围，并且它们都受代码长度的限制。

(3) 在值的可表示范围内，整数一定是精确表示，而实数的表示可能只是近似的。例如，前面提到过的0.9的机内表示，不管尾数定多少位，始终只能是近似表示。

1.3 程序和程序设计语言

计算机是一种强有力的工具，能以极高的速度对存放在主存储器里的数据进行处理，并获得期望的输出结果。原始数据的输入、处理到结果数据的输出，都是由人事先编好程序，然后将程序和原始数据一同输入到计算机里，交由计算机自动完成的。所谓程序就是指令的序列，它描述了需要处理的数据以及对这些数据需要执行的操作。

最原始的指令是机器指令。它由计算机的设计者设计，作为计算机线路的一部分而建立在机器内部。典型的计算机可以有几十或上百种机器指令（计算机的全部指令称为该计算机的指令系统），每种指令完成某种特定的操作功能。典型的机器指令格式是：

操作码	地址码
-----	-----

其中：操作码指明操作属性，如加、减、乘、除、存取、传送、移位、比较、转移、输入、输出等，地址码给出参与运算的对象所在的位置。操作码和地址码都是由0和1组成的二进制代码，代表了它们的编号。根据机器提供的指令系统所编制出来的程序叫机器指令程序。机器指令程序输入到主存储器后，一经启动便可直接运行得到结果。机器指令程序具有效率高、速度快的特点。然而，用机器指令编制程序是件非常繁重的工作，不但要熟悉各种指令代码，而且要把一个复杂的计算问题分解成能由单个指令组合在一起完成。用机器指令写出来的程序难读、难调试、难于保证程序的正确性，并且缺乏通用性。因为，不同的计算机有不同的指令系统，在某种机型上能正确运行的机器指令程序，在另一机型上完全行不通。

为了绕开机器指令，克服机器指令程序的缺陷，人们提出了程序设计语言的构想。所谓程序设计语言是用来描述计算过程的一种符号系统。有了程序设计语言，就可甩开机器指令而使用我们熟悉、习惯的语言符号来编写程序，这是程序设计的重大突破。过去30年中，随着计算机硬件的发展，人们创造了许多介于英语和机器指令之间的各种程序设计语言。如果从语言的级别来分，则大致可分为两种：汇编语言（低级）和高级语言。

汇编语言是一种面向机器的程序设计语言。它的特点是用便于记忆的符号来代替机器指令的操作码和地址码,因此,相对机器指令它便于理解和记忆。早期的汇编语言,基本上是一条汇编指令对应一条机器指令,仍然是和机器密切相关,编程工作量大,无通用性,所以人们常把它和机器指令一起划属于低级语言。

高级语言有点象自然语言,由专门的符号根据词汇规则构成单词,由单词根据语法规则构成语句,每种语句有确切的语义并且能由计算机解释。高级语言包含许多英语单词,有“自然化”的特点;高级语言书写计算式子接近于熟知的书写数学公式的规则。高级语言已和原始的机器指令完全分离,一条高级语言指令(即语句)常常相当于几条或几十条机器指令。因此,高级语言的出现,给程序设计从形式和内容上均带来了重大的变更,大大提高了程序的生产效率。

从1956年FORTRAN语言问世,至今已有数百种高级语言。从使用角度来看,可分为通用程序设计语言与专用程序设计语言。在通用程序设计语言中,又可分为科学计算用语、数据处理语言和实时语言等。目前国内外比较通用的高级语言有:

BASIC	简单易学的会话式语言;
FORTRAN	面向科学计算的语言;
COBOL	面向商业数据处理的语言;
PL/1	功能较强的通用语言;
PASCAL	结构程序设计语言;
C	既具有高级数据结构和控制结构,又具有低级语言特色的通用语言;
LISP	符号处理语言;
PROLOG	逻辑型语言;
ADA	美国国防部的标准高级语言。

不管是高级语言还是汇编语言程序,均不能由计算机直接执行。计算机能直接执行的只有机器指令程序。我们把用高级语言或汇编语言书写的程序叫源程序。源程序必须经过一个“翻译程序”把它翻译成机器指令程序后才能被执行。我们把翻译后的机器指令程序叫目标程序。翻译汇编语言的程序叫汇编程序,翻译高级语言的程序叫编译程序。对于任何计算机,若要它能识别某种高级语言,就必须为它配备一个该种高级语言的编译程序。编译程序一般均由计算机厂家组织研制,并和计算机硬件一起提供给用户。例如,若要使你手边的计算机能运行PASCAL程序,就必须安装有PASCAL编译程序,它的功能是把用PASCAL语言书写的源程序翻译成等价的目标程序,然后运行目标程序才得到你所期望的结果。

1.4 PASCAL程序设计语言概述

1.4.1 PASCAL程序设计语言的特点

PASCAL程序设计语言是沃思(N. Wirth)教授于60年代末在瑞士苏黎世联邦工业大学创立的。第一个PASCAL编译程序于1970年实现,翌年公布了PASCAL语言的最初版本。1974年发表了PASCAL用户手册和修改报告,以后人们把这个修改报告称为标准PASCAL。

PASCAL语言是按照结构程序设计原则设计的一种通用程序设计语言。沃思教授设计

PASCAL语言的最初目的是：第一，创建一个适合于程序设计教学的语言，通过这门语言课的教学使计算机科学中的一些基本概念受到系统训练；第二，提供一些令人满意的整体特性，它们能够编译成高效率的代码。PASCAL问世后的十多年实践证明，沃思教授的目的不仅已经完全达到，而且取得了极大的成功。PASCAL不依靠任何组织或个人的推广，迅速地在全世界范围内得到重视，不仅大学里将它作为讲述结构程序设计、培养学生良好程序设计风格的典范，而且它也日益广泛地在商业、管理、控制等应用领域受到欢迎。

PASCAL的成功在于它的以下特色：

(1) PASCAL具有丰富的数据类型，有着象枚举、子界、数组、记录、集合、文件、指针等众多的用户自定义数据类型，能够用来描述复杂的数据对象，十分便于书写系统程序和应用程序。

(2) PASCAL提供的语言设施体现了结构程序设计的原则，有着简明通用的语句，基本结构少，但框架优美，功能很强；算法和数据结构采用分层构造，可自然地应用自顶向下的程序设计技术；程序可读性好，编译简单，目标代码效率较高。

1.4.2 PASCAL程序的结构

PASCAL语言的创始人沃思教授提出一个著名的公式：

算法 + 数据结构 = 程序

无疑，这是对程序内涵所作的精辟论断。公式告诉我们，程序由两部分组成：一部分是关于算法的描述，算法指明对数据要实现的操作，它由一系列的语句组成；另一部分是关于数据结构的描述，数据结构说明数据的逻辑结构和物理结构，它由一系列的说明和定义来描述。对一个实际问题进行程序设计时，通常是先确定合适的数据结构，然后建立相应的算法。算法和数据结构是密切相关的。

下面是一个简单的程序实例。程序的工作过程是：提示输入圆的半径R，当用户从键盘打入R的数值后，计算圆的周长C和面积S，然后将C和S的值分别输出。我们通过这一实例来说明PASCAL程序的一般结构。（为书写、阅读和输入的方便，本书中文字叙述或图表中常用小写字母表示的变量，而在程序中均用相应的大写字母替代。）

```
PROGRAM CIRCLE (INPUT, OUTPUT);
  (*A program for calculating the circumference and area of a circle*)
  CONST PI = 3.14159; (*constant definition*)
  VAR R, D, S: REAL; (*variable declaration*)
  BEGIN (*part of statements*)
    WRITE ('enter radius of a circle: ');
    READLN(R);
    C = 2 * PI * R;
    S = PI * R * R;
    WRITELN('circumference = ', C);
    WRITELN('area = ', S);
  END.
```

从程序的词法成分来看，标准PASCAL程序的基本词汇由英文字母符、数字符、保留字和特殊符号组成。特殊符号有 + - * / = < > () [] . , : ; ' { } < > < = > = : = . . ^ 共24个。PASCAL程序中出现的所有单词，或本身就是基本词汇，或可由