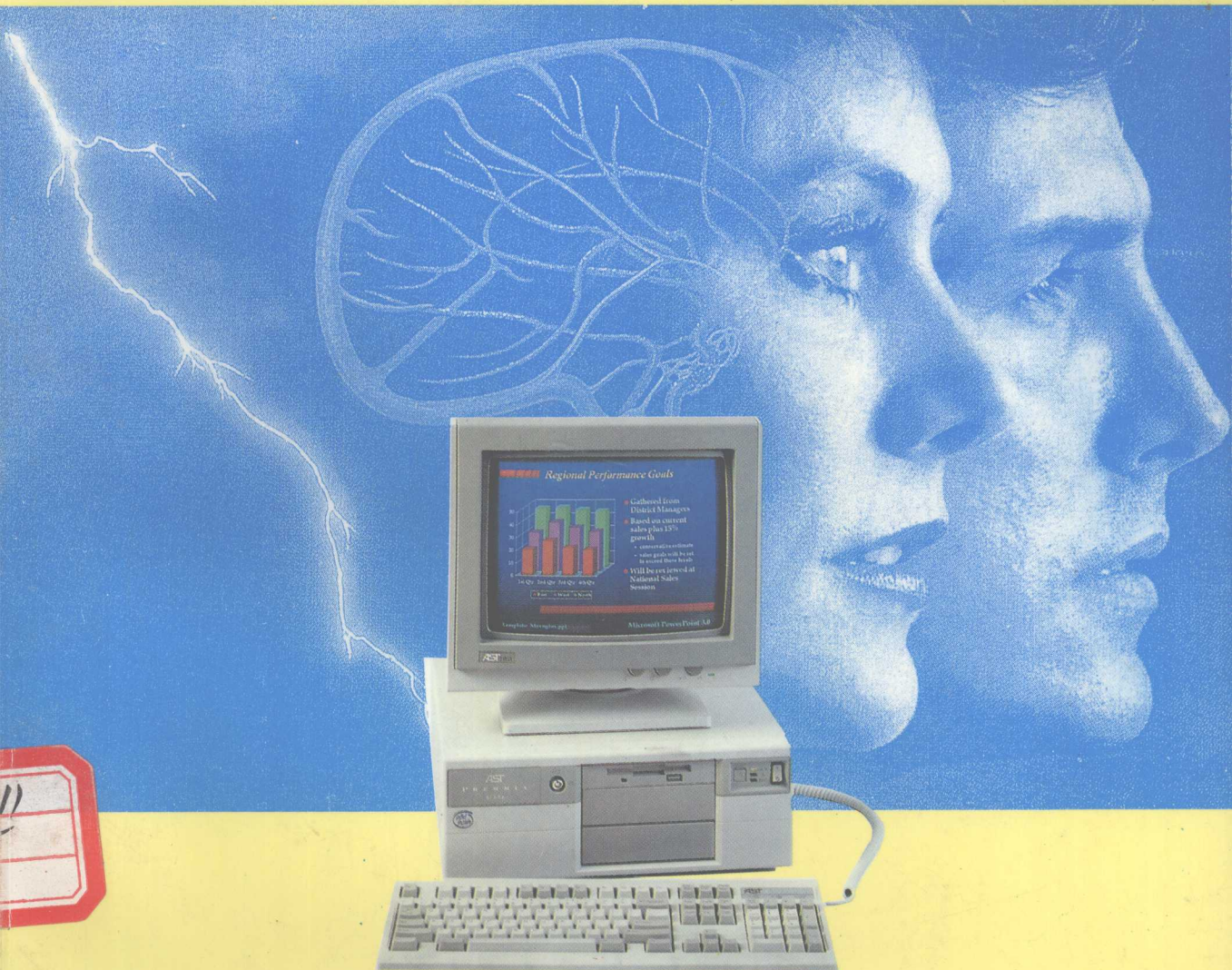


郑国梁 主编

GAOJI YUYIAN CHENGXU SHEJI

全国高等教育自学考试教材  
计算机及其应用专业(专科)

# 高级语言 程序设计



南京大学出版社

全国高等教育自学考试教材

计算机及其应用专业

# 高级语言程序设计

郑国梁 主 编

胡学联 副主编

郭肇德 主 审

南京大学出版社

## 内 容 简 介

本书全面地介绍了PASCAL语言的各种成份,并注重程序设计方法与技巧的训练。全书共有十四章,分为两大部分,前八章为基础部分,后六章为提高部分。

本书叙述通俗易懂,内容循序渐进,示例丰富,便于自学,习题难易适中。其深度和范围完全符合全国高等教育自学考试高级语言程序设计自学考试大纲要求。

本书可作为高等教育自学考试计算机及其应用专业教材,同时可供该专业各类本科或专科班作为教材或教学参考书,对从事计算机系统软件及应用软件开发的工程技术人员,也是一本较好的参考书。

(苏)新登字第 011 号

### 高级语言程序设计

GAOJI YUYAN CHENGXU SHEJI

郑国梁 主 编

胡学联 副主编

南京大学出版社出版发行

(南京大学校内 邮政编码:210008)

中国科学技术大学印刷厂印刷

开本 787×1092/16 印张 6 字数:384千

1994年2月第1版 1995年12月第2次印刷

印 35001-85000

ISBN 7-305-02594-1/TP·94

定价:18.00元

## 出版前言

高等教育自学考试教材是高等教育自学考试工作的一项基本建设。经国家教育委员会同意，我们拟有计划、有步骤地组织编写一些高等教育自学考试教材，以满足社会自学和适应考试的需要。《高级语言程序设计》是为高等教育自学考试计算机及其应用专业组编的一套教材中的一种。这本教材根据专业考试计划，从造就和选拔人才的需要出发，按照全国颁布的《高等教育自学考试计算机及其应用专业（专科）高级语言程序设计自学考试大纲》的要求，结合自学考试的特点，组织高等院校一些专家学者集体编写而成的。

计算机及其应用专业《高级语言程序设计》自学考试教材，是供个人自学、社会助学和国家考试使用的。现经组织专家审定同意予以出版发行。我们相信，随着高教自学考试教材的陆续出版，必将对我国高等教育事业的发展，保证自学考试的质量起到积极的促进作用。

编写高等教育自学考试教材是一种新的尝试，希望得到社会各方面的关怀和支持，使它在使用中不断提高和日臻完善。

全国高等教育自学考试指导委员会

一九九四年元月

## 编者的话

PASCAL 语言是在世界上广泛使用的高级语言之一。PASCAL 语言系统地体现了结构程序设计的思想,它能清晰地、分层地表达数据结构和算法。该语言简明,易学。用该语言写出来的程序易读性好,便于查找和纠正错误。该语言适用范围广泛,在科技界、教育界、企业界倍受推崇。现有的计算机系统,无论是大型、中型还是小型、微型,都配有该语言。

在全国高等教育自学考试计算机及其应用专业的考试计划中,指定高级语言程序设计课程选用 PASCAL 语言,并且制定了相应的自学考试大纲。本书便是根据这个大纲编写而成的。

本书的目的有二:第一,用作以程序设计语言 PASCAL 为基础的程序设计教程,教会读者编写良好的程序。第二,作为 PASCAL 语言的导引,采取循序渐进的方式逐步引导读者掌握 PASCAL 语言的基本内容。

本书共十四章,可分为基础部分(前八章)和提高部分(后六章)。基础部分详细地介绍 PASCAL 语言中的基本数据类型,几乎所有的语句及过程与函数,并讨论了程序设计方法,使读者可以编写一般的程序,解决一些基本的问题。提高部分介绍 PASCAL 语言程序设计中其余深一些的内容,掌握这一部分便可以用 PASCAL 语言解比较复杂的问题。书中打 \* 号的内容以及与此有关的部分具有一定难度,可供有兴趣的读者进一步学习之用。

获得程序设计的系统方法,较好地掌握一种语言,除了阅读书本以外,不可缺少的另一重要环节是上机实践。从这个意义上来说,高级语言程序设计课程是一门基础性、实践性都很强的课程。为此,本书提供了较多的例题和习题,读者可以直接拿来作为上机练习题,并且尝试改进它们。

本书承蒙南京通讯工程学院郭肇德教授、南京航空学院徐宝文副教授、南京大学钱士钧副教授审阅,并且提出了许多宝贵意见,在此谨表谢意;庄昌瓚、陈悦、顾薇薇、白捷以及中国计算机函授学院激光照排室全体同志为本书的录入、排版付出了辛勤的劳动,在此一并致谢。

本书由郑国梁编写第九章至第十四章,胡学联编写第一章至第三章、第六章及附录,潘金贵编写第七章和第八章,陈晓红编写第四章、第五章。鉴于水平所限,不妥之处在所难免,欢迎读者批评指正。

作 者

一九九四年元月

# 目 录

<b>第一章 概述</b> .....	(1)
§ 1.1 计算机硬件与软件 .....	(1)
§ 1.2 程序和程序设计语言 .....	(4)
§ 1.3 问题求解 .....	(7)
习题 .....	(10)
<b>第二章 程序结构与语法描述</b> .....	(11)
§ 2.1 PASCAL 语言词汇集 .....	(11)
§ 2.2 程序结构 .....	(14)
§ 2.3 语法描述 .....	(17)
习题 .....	(20)
<b>第三章 简单数据与类型</b> .....	(21)
§ 3.1 常量和常量定义 .....	(21)
§ 3.2 变量和变量说明 .....	(24)
§ 3.3 类型和类型定义 .....	(25)
§ 3.4 简单类型 .....	(27)
习题 .....	(35)
<b>第四章 表达式</b> .....	(37)
§ 4.1 表达式的组成 .....	(37)
§ 4.2 标准函数 .....	(41)
§ 4.3 表达式的类型及类型一致性 .....	(43)
习题 .....	(45)
<b>第五章 语句</b> .....	(46)
§ 5.1 语句的作用及其分类 .....	(46)
§ 5.2 赋值语句 .....	(47)
§ 5.3 输入/输出语句 .....	(48)
§ 5.4 复合语句 .....	(54)
§ 5.5 条件语句 .....	(55)
§ 5.6 重复语句 .....	(62)
* § 5.7 转向语句 .....	(72)
§ 5.8 空语句 .....	(74)
习题 .....	(77)
<b>第六章 过程与函数(I)</b> .....	(80)
§ 6.1 过程与函数的提出 .....	(80)
§ 6.2 函数 .....	(81)
§ 6.3 过程 .....	(85)
§ 6.4 参数一致性及其选择 .....	(86)
§ 6.5 应用举例 .....	(88)
习题 .....	(92)
<b>第七章 程序的层次及标识符的作用域</b> .....	(95)
§ 7.1 程序的层次结构 .....	(95)
§ 7.2 标识符的定义性出现和应用性出现 .....	(97)
§ 7.3 标识符的作用域 .....	(98)
§ 7.4 标识符的局部性与非局部性 .....	(101)
习题 .....	(103)

<b>第八章 程序设计方法</b> .....	(107)
§ 8.1 程序设计风格 .....	(107)
§ 8.2 结构化程序设计 .....	(110)
§ 8.3 结构化程序设计示例 .....	(113)
§ 8.4 程序的调试与测试 .....	(127)
习题 .....	(133)
<b>第九章 数组类型</b> .....	(135)
§ 9.1 构造类型 .....	(135)
§ 9.2 一维数组类型 .....	(136)
§ 9.3 一维数组的应用 .....	(138)
§ 9.4 多维数组 .....	(144)
* § 9.5 紧缩数组类型 .....	(150)
§ 9.6 字符串 .....	(151)
习题 .....	(153)
<b>第十章 记录类型</b> .....	(156)
§ 10.1 记录类型 .....	(156)
§ 10.2 记录变量和域命名符 .....	(158)
§ 10.3 开域语句 .....	(161)
* § 10.4 记录的变体部分 .....	(165)
习题 .....	(170)
<b>第十一章 集合类型</b> .....	(172)
§ 11.1 集合类型 .....	(172)
§ 11.2 集合构造符 .....	(174)
§ 11.3 集合运算 .....	(174)
§ 11.4 类型相容与赋值相容 .....	(179)
习题 .....	(180)
<b>第十二章 文件类型</b> .....	(182)
§ 12.1 文件概念 .....	(182)
§ 12.2 文件类型、文件变量、缓冲区变量 .....	(183)
§ 12.3 文件读写 .....	(184)
§ 12.4 内部文件与外部文件 .....	(191)
§ 12.5 正文文件 .....	(192)
§ 12.6 标准文件 INPUT, OUTPUT .....	(195)
* § 12.7 READ 和 WRITE 的特殊功能 .....	(197)
习题 .....	(201)
<b>第十三章 动态数据结构</b> .....	(203)
§ 13.1 动态数据概念 .....	(203)
§ 13.2 指针类型 .....	(204)
§ 13.3 指针变量及标识变量 .....	(205)
§ 13.4 动态变量的产生和撤消 .....	(207)
§ 13.5 动态变量的应用 .....	(211)
习题 .....	(227)
<b>第十四章 过程与函数(I)</b> .....	(229)
§ 14.1 递归 .....	(229)
§ 14.2 向前引用和指示字 .....	(237)
§ 14.3 函数副作用 .....	(238)
习题 .....	(239)
<b>附录 I PASCAL 语言语法图</b> .....	(242)
<b>附录 II PASCAL 语言词汇表</b> .....	(248)
<b>附录 III ASCII 字符集</b> .....	(249)

# 第一章

## 概 述

**【内容提要】** 本章简要介绍了计算机系统的基本组成,对程序和程序设计语言等术语作了概要说明和解释,并且指出了程序设计语言的两个侧面:语法和语义,以及程序设计语言一般包含的四个基本部份。最后给出了利用计算机系统解题应遵循的几个步骤。

### § 1.1 计算机硬件与软件

一个计算机系统通常包含两大部分:硬件和软件。硬件是指系统中的实际设备,它可以是电子的、电的、磁的、机械的、光学的元件或设备,或由它们组成的计算机部件或计算机。软件,则是相对于硬件而言的,它包括运行机器所需的各种程序和相关文件资料。

#### 一、硬件

一般说来,一台计算机有四大功能部件:中央处理器(CPU),存储器,输入设备和输出设备。图 1-1 给出了计算机的基本组成框图。

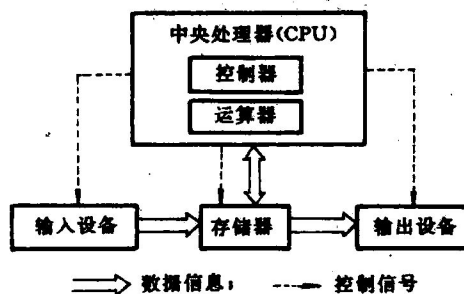


图 1-1 计算机基本组成框图



## 1. 中央处理器

中央处理器包含用于解释和执行指令的线路,以及为执行指令所必需的运算、逻辑和控制线路。中央处理器包括运算器和控制器,它们是计算系统的两个基本部件。

控制器被用于控制指令执行的顺序。它从存储器取来指令并进行解释,根据译码将相应的控制信号送给运算器或其它功能部件。

运算器又称算术和逻辑部件,用于完成算术运算或逻辑运算。控制器提供给它适当的数据并使它执行指定的运算。

## 2. 存储器

存储器用来接收和保存数据,而且能根据命令提供这些数据。存储器的特点是,向存储器中某一指定位置写入一个信息后,可以反复从这个存储位置读出这个信息,直到再次写入另一个信息为止。计算机系统通常有两类存储器:主存储器和辅助存储器。主存储器是计算机的主要工作存储器,对它可以直接进行访问。在存储器中,每个存储位置都对应一个存储地址,通过这种地址可以向存储器写信息或从存储器读信息。辅助存储器通常是指磁盘、磁带、磁鼓等,它是为了弥补主存储器容量不足而附加的一种存储器。在CPU的控制下,它可以和主存储器自动成批地交换数据。

## 3. 输入和输出设备

输入和输出设备分别用来把信息读入计算机和把信息从计算机中读出。

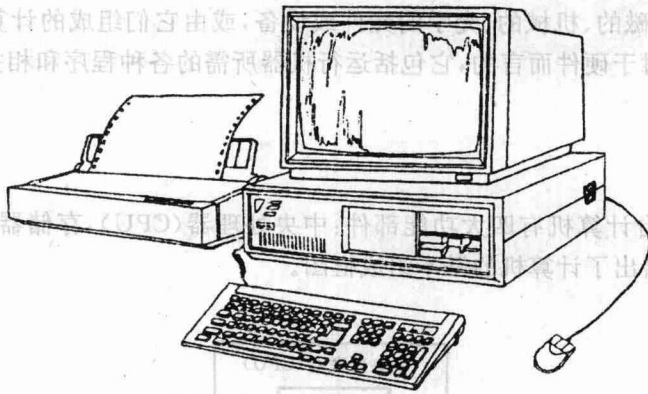


图 1-2 微型计算机外观图

例如,程序和数据可以通过卡片或纸带读入计算机,就需要相应的卡片读入机或光电输入机这类输入设备。在计算机系统的终端或在微型计算机上,可以用键盘直接将程序和输入数据输入计算机,键盘就是输入设备。

输出设备可以是行式打印机、控制台打印机、终端屏幕、卡片穿孔机或纸带穿孔机等。

一台计算机一般都要配有一台输入设备和一台输出设备,也可以配有多台。对于微型计算机来说常见的是配有一台键盘和一台屏幕显示器。

每种计算机,都能提供一组基本功能,说到底就是,它只能识别和执行一些基本的命令,比如两数相加、减,从输入设备上读一个数等。我们称这些基本功能为指令。这些基本指令的集合称作机器语言。

在计算机中,所有的指令和数据都是用二进制表示的。程序人员直接同这些指令和数据打交道十分烦琐,而且极易出错,从而极大地影响到计算机的推广应用。解决这一问题的途径是发展软件。

## 二、软件

在计算机系统中,硬件是物质基础、是躯体,软件则是指挥枢纽、是灵魂。

软件一词有三层含义:

- ①指程序和有关的文件及数据;
- ②指特定计算机系统中所有前述含义下的软件的整体;
- ③指为了研究、开发以及维护前述含义下的软件所涉及的理论、原则及技术所构成的学科,在这种意义下,应称为软件学,简称软件。

软件的作用是:

- ①用户和硬件的界面,用户一般通过软件来和机器交往;
- ②计算机系统的指挥机构;
- ③计算机系统体系设计的重要依据。

软件的发展阶段:

第一阶段(1945~1955)。其特点是串行工作方式。用户采用低级语言(即前述的机器语言或和机器语言接近的汇编语言)编制程序。

第二阶段(1956~1967)。其特点是多道并行方式。用户一般采用高级语言编写程序。

第三阶段(1968~迄今)。其特点是软件工程。

在六十年代中期,由于软件的复杂程度高,研制周期长,正确性难保证,出现了人们难于控制软件发展的局面,即所谓软件危机。造成这种危机的重要原因是,软件开发方式落后,大都是个体技艺、手工作坊,这和大型软件的开发不相适应,因为软件开发本身是一项工程,因此,必须寻求用工程方法来开发与维护软件,从而,在1968年出现了“软件工程”。

一台计算机总是由硬件和软件两大部分组成的。硬件是可见的实体,它一经确定后是不轻易更改的;软件则是依附硬件而存在的,随着硬件环境、应用领域或范围的变化,它需不断开发维护,经常处于改变状态之中。如果没有硬件,便履行不了基本功能,那末软件也就失去了效用;但仅有硬件,没有软件,计算机也不能发挥它的潜在能力。

## § 1.2 程序和程序设计语言

### 一、程序

程序是对所要解决问题的各个对象和处理规则的描述。或者说是为解决某一问题而设计的一系列指令(由操作和操作对象组成)。

实现上述描述任务的过程就是程序设计。

[例 1-1] 要求在终端上显示或打印输出一句话：“我们是中国计算机函授学院的学生”。

分析：解决这一问题，需要按计算机系统所能理解的记号和规则进行程序设计。

程序：

```
program example (output);
  (* this is a simple example *)
begin
  writeln (' we are students of computer correspondence college of China' )
end.
```

其中 program 为界限符号，表示程序从此开始；example 为程序标识符，它表示该程序的名字；用 begin 和 end 括起来的是一个语句，它表示该程序的可执行部分。

程序编制完成后，便可利用输入设备将程序录入到计算机系统中。在微型机系统中，通常是利用键盘逐字逐段地键入到系统中，同时在终端显示器上显示出已键入的部分。

录入过程中如有误操作，可以及时修改。录入完毕，再向计算机系统发出请求处理的命令。经过计算机系统的工作，在屏幕上将显示(或在打印机上印出)：

```
we are students of computer correspondence college of china.
```

[例 1-2] 按输入的半径值，计算圆的周长和面积。

分析：如果半径为  $r$ ，则圆周长为  $2\pi r$ ，圆面积为  $\pi r^2$

程序：

```
program calculation (input,output);
  (* 本程序用以计算圆周长和圆面积 *)
const pi=3.1415926535;
var radius, circumference,area,real;
begin
  read(radius); (* 读入半径值 *)
  circumference:=2 * pi * radius; (* 计算圆周长 *)
  area:=pi * radius * radius; (* 计算圆面积 *)
  writeln('The radius of circle is ', radius,11:9);
  writeln ('The circumference of circle is ', circumference,11:9);
  writeln ('The area of circle is ', area,11:9)
end.
```

以 const 起首的一行为常量定义,它定义了一个以 pi 为名字的常量,它的值为 3.1415926535;以 var 起首的一行为变量说明,它说明了三个变量,即 radius, circumference, area,三者类型均为实型(real)。此例中用 begin 和 end 括起来的有六个语句。第一个是读语句,其作用是把输入介质上的当前值读入到分配给 radius 的存储位置,作为 radius 的当前值。接下去两行是两个赋值语句,分别用以计算圆周长和圆面积。末三行是三个写语句,其作用是,把半径、圆周长、圆面积分别打印出来。在这个程序中,常量说明与变量说明刻划计算对象;语句则刻划计算规则。

执行该程序时,需输入一个半径的值,若半径长是 2.1,则输入 2.1 后,计算机系统便显示:

```
The radius of circle is 2.10000000  
The circumference of circle is 13.194689145  
The area of circle is 13.854423602
```

需要说明的是,在上述两例子的程序中,多处出现用{与}或(\*与\*)括起的部分,这是程序的注释部分。注释部分一般是英文字符,在中西文兼容的系统也可以是中文。它是为改善程序的可读性和可理解性而插入的,对程序的正确性和运行都不会有影响。本书后面的例子中,注释部分将采取中西文混用的方式。

## 二、程序设计语言

程序设计语言是用来编写计算机程序的语言。语言的基础是一组记号与一组规则。根据规则由记号构成的记号串就是程序。

在考察程序设计语言时,应该注意它的两个侧面:语法和语义。语法表示程序的构成规则,语义表示程序的含义。

语言的好坏不仅关系到使用是否方便,而且影响到程序人员所写出的程序的质量。

语言的发展是从低级语言到高级语言。低级语言一般指机器语言或与之接近的汇编语言。高级语言指的是像 PASCAL 这类(原则上)与具体机器无关的语言。

语言的种类千差万别,但是一般说来,其成份有如下四种:

- ①数据成分。用以描述程序中所涉及的数据;
- ②运算成分。用以描述程序中所包含的运算;
- ③控制成分。用以表达程序中的控制结构;
- ④传输成分。用以表达程序中的数据传输。

到目前为止,常用的语言有一百多种,但对每个程序人员来说,最常用的也只有少数几种。通常所用的语言,早期有 FORTRAN、ALGOL、COBOL;七十年代流行的是 PASCAL、C;八十年代的代表作是 Modula-2、Ada,以及逻辑式程序设计语言 PROLOG、面向对象的程序设计语言 smalltalk 等等。

## 三、程序的执行

程序设计语言从低级语言(机器语言、汇编语言)向高级语言的发展,极大地提高了人

们编写程序的效率,改善了程序的易读性,也使程序的可靠性和正确性有了相应保证。

高级语言,例如 PASCAL 语言,与低级语言相比,从程序设计的角度来看主要有两个优点:第一,PASCAL 语言比机器语言更接近于人自己的语言,因此用它易于编写程序;第二,PASCAL 程序具有较好的易移植性,这主要是指在一台机器上运行的 PASCAL 程序,通常可以在许多不同类型的机器上运行,或者只要稍加修改即可运行。而用机器语言编写的程序通常只能在特定的机器上运行。

计算机一般不能直接执行高级语言程序,它们只能执行机器语言程序。因此需要将 PASCAL 一类的高级语言程序翻译成计算机所能理解的语言,即机器语言。而这个翻译过程仍然可以交给计算机自己去进行,担任这个翻译工作的程序称为编译程序(或解译程序)。PASCAL 语言的编译程序接受由 PASCAL 语言编写的程序——源程序,并将它翻译成等价的机器语言程序——目标程序。此后该目标程序就被保存在计算机的存储设备中,以便执行。其过程如图 1-3 所示。

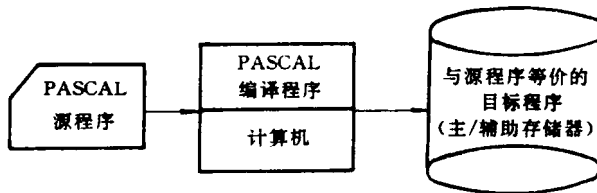


图 1-3 准备执行 PASCAL 源程序

执行目标程序时,将根据需要把数据输给对应的数据对象,把运行结果按指定的方式输出来。图 1-4 表示例 1-2 计算圆周长和圆面积程序中的输入和输出,以及执行程序时的信息流。

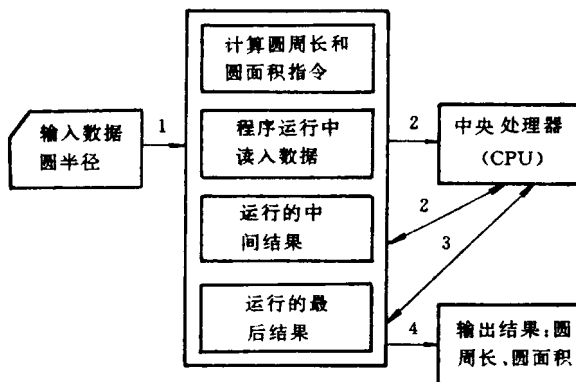


图 1-4 执行程序时的信息流

## § 1.3 问题求解

### 一、求解过程

通过计算机系统来求解问题，一般需经过如下几步：

- ①明确问题要求。
- ②建立数学模型确定计算方法。
- ③算法分析。确定待解问题的数据对象和求解步骤，或者说确定数据与算法。
- ④程序设计。使用程序设计语言把由(3)确定的数据和算法根据自顶向下、逐步求精的原则描述出来，逐步形成符合语法规则的程序。
- ⑤上机计算。上机计算包括源程序准备、编译与运行。源程序准备指的是将编写好的源程序在输入介质上准备好，编译程序接受源程序，并将源程序翻译成目标程序。请求计算机系统运行程序后，计算机系统便进行指定的求解，有时还要读入准备好的数据，并显示或输出计算结果。整个流程如图 1-5 所示。

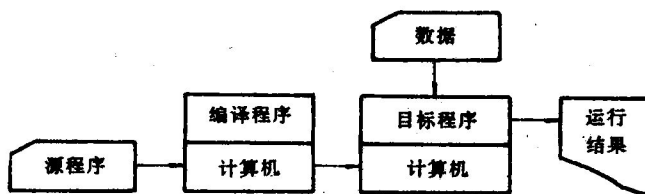


图 1-5 计算流程图(一)

上机计算时，常常会有错误发生，这些错误通常来自两个方面，一类是书写或录入程序时的疏忽和遗漏，使程序不合语法要求，计算机系统无法理解。如缺少标点符号，缺少语句成分等等。这类错误在编译阶段都能检查出来。编译程序将这些错误情况反馈给程序人员，程序人员修改源程序后重新上机计算。另一类错误，往往是算法上的错误。例如，没有给出变量的初始值，计算公式本身有遗漏或错误，出现了不应有的运算，或者计算机不能完成的计算，例如除法的除数是 0 等等。这些一般是在运行阶段才能发现，甚至没被发现而得到错误的结果。这一类错误的纠正往往比较麻烦。但是，一般可以通过调试解决。另一方面，为了确信计算结果的正确，还可根据问题预先知道的一些结果进行测试，即用预先知道的数据，在一些程序位置上，检验是否得到预期的结果。调试和测试都能在一定程度上提供程序正确与否的信息。这些调试信息和测试信息将指导修改原来的算法和源程序，然后再重新进行编译和运行，直到得出正确结果为止。这个流程图可由图 1-6 表示。

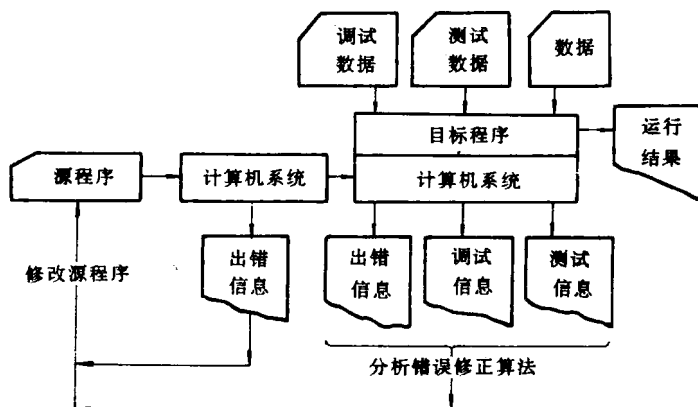


图 1-6 计算机流程图(二)

## 二、文件规范 \*

为了便于程序特别是中、大型程序的管理,加快程序形成,便于程序的推广与维护,针对求解问题的流程,应该强调在每一工作步骤形成一个规范的文件,一般应该形成五个文件。

①问题要求 指待解问题的各项要求、指标,应当十分明确和详细地书写。这既是以后设计的依据,也是检验程序正确性的唯一标准。

②算法分析 包括待解问题的数学模型、计算方法,指出求解该问题要使用哪些数据,这些数据的要求形式以及怎样对数据加工的算法。

③程序设计 要自顶向下、逐步求精地产生程序。先描述总体算法,再描述局部细节。程序应有注解,并有详细使用说明。

④调试报告 在调试过程中所出现的问题和解决的措施,以及采用哪些数据验证程序,结果和实际是否相符等方面应有的记载和分析。

⑤小结 通过求解这一问题,分析采用的算法有哪些特点,这一程序还能适用于哪些方面,以及怎样扩充程序功能等。

这五个文件应当在整个求解问题的过程中逐步形成,而不是在调试完毕后再形成。应该要求在一开始就有这种工作习惯。

除了在求解问题过程中逐步形成的文件外,当程序交付使用时,一般应有两个不同的文件,即用户文件和技术文件。

用户文件一般包含下列内容:

- ①程序名;
- ②对程序简短的、非技术性的描述;
- ③程序需要输入的数据,包括这些数据的范围、格式以及从何种设备输入;
- ④当接收有效的数据后,程序的正常输出;
- ⑤异常报告;

- ⑥程序的限制；
- ⑦在特定机器上运行所必须的命令序列；
- ⑧程序编写者的姓名、地址。

技术文件一般包含下列内容：

- ①程序名及其目的；
- ②程序研制的历史及现状(包括程序编写者姓名、地址)；
- ③程序总的结构；
- ④每个模块的说明；
- ⑤关键性的数据结构说明；
- ⑥调试和测试措施。

## 小 结

1. 计算机系统是计算机硬件和软件的总称。硬件是指系统中的实际设备；软件是指运行机器所需的各种程序和相关的资料。

2. 计算机硬件包括四个功能部件：中央处理器(CPU)、存储器、输入设备和输出设备。中央处理器是指包含指令的解释和执行线路以及为执行指令所必需的运算和控制线路；存储器用来接收和保存数据，并能根据命令提供这些数据；输入和输出设备用来将数据送入计算机和将数据从计算机内送出。

3. 计算机软件有三层含义：

- ①指程序及有关文件；
- ②指特定计算机系统中所有前述含义下的软件的整体；
- ③指为了研究、开发以及维护前述含义下的软件所涉及的理论、原则及技术所构成的学科。

4. 软件的作用包括三个方面：

- ①作为用户和硬件的界面；
- ②计算机系统的指挥机构；
- ③是计算机系统体系结构设计的重要依据；

5. 程序是对所要解决问题的各个对象和处理规则的描述；程序设计则是编制、调试、执行程序的过程。

6. 程序设计语言是用来书写计算机程序的语言；语言的基础是一组记号与一组规则；根据规则由记号构成的记号串的总称就是语言；在程序设计语言中这些记号串就是程序。

7. 一个程序设计语言包括两个方面，即语法、语义。语法表示程序的构成规则，亦即表示构成语言的各个记号之间的组合规律，但不涉及这些记号的特定含义；语义表示程序的含义，亦即表示按照各种方法所表示的各个记号的特定含义。

8. 一个程序设计语言一般包括四种成分：

- ①数据成分：用以描述程序中所涉及的数据；



- ②运算成分:用以描述程序中所包含的运算;
  - ③控制成分:用以表达程序中的控制构造;
  - ④传输成分:用以表达程序中的数据传输。
9. 利用计算机系统求解问题一般分为如下几步:
- ①明确问题要求;
  - ②提出求解的数学模型和计算方法;
  - ③算法分析,确定数据结构与算法;
  - ④程序设计;
  - ⑤上机计算。

## 习 题

1. 计算机硬件由哪几部分组成?
2. 软件的作用是什么?
3. 什么叫程序? 什么叫程序设计?
4. 程序设计语言一般包含哪几个主要成份?
5. 编译程序的功能是什么?
6. 在计算机上解题,通常需要经过哪几个阶段?
7. 什么是源程序,什么是目标程序?
8. 程序中一般会出现哪几类错误? 可分别在什么时候发现?