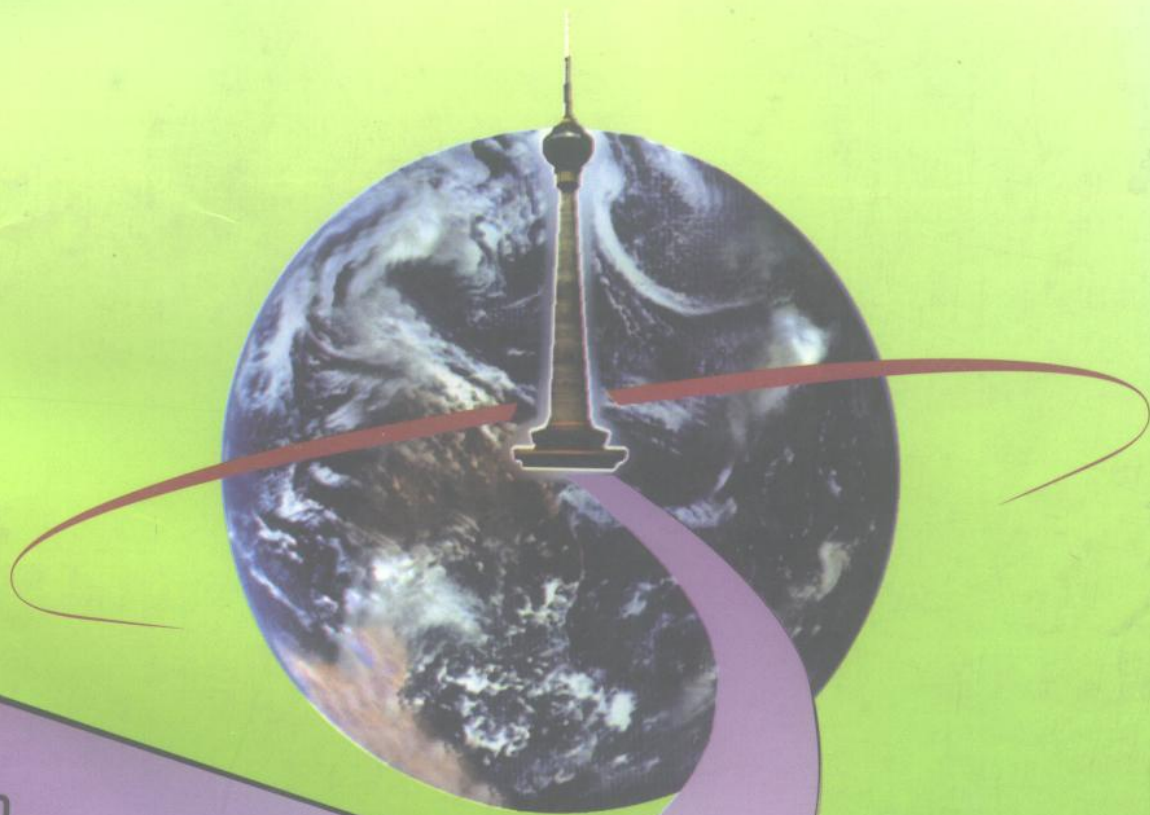




全国非计算机专业等级考试、
自学考试辅导丛书

新编 PASCAL 语言程序设计 自学辅导

本丛书编委会



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

新编 PASCAL 语言程序设计

自学辅导

本丛书编委会

电子工业出版社

内 容 提 要

本书是全国非计算机专业等级考试、自学考试辅导丛书的第五册，全书以通俗、浅显的文字介绍了PASCAL语言的基础知识和程序设计技术。全书突出基本技能的培养，并配有作业及答案。书中收集了初学者易出的错误，并给予答疑辅导。本书适合于参加各类计算机等级考试的读者自学使用，亦可作为计算机基础教育的入门教材。

全国非计算机专业等级考试、自学考试辅导丛书

新编 PASCAL 语言程序设计自学辅导

本丛书编委会

责任编辑：郭立 贾贺

*

电子工业出版社出版

北京市海淀区万寿路 173 信箱(100036)

电子工业出版社发行 各地新华书店经销

中国科学院印刷厂印刷

*

开本 787×1092 毫米 1/16 印张：17 字数：403 千字

1996 年 6 月第一版 1996 年 6 月北京第一次印刷

印数：00001—10100 册 定价：18.50 元

ISBN 7-5053-3649-5/TP·1509

全国非计算机专业等级考试、自学考试辅导丛书编委会

顾 问 刘乃琦

策 划 王明君

编 委 许 远 何成彦 吴 跃

陈文字 陈周坤

前 言

随着科学技术的迅猛发展,计算机已成为各个学科领域不可缺少的应用工具,计算机知识和应用能力已成为当代大学生知识和能力结构的一个重要组成部分,也是我国教育培养跨世纪人才最突出的需要加强的环节之一。目前在高校中普遍开展的计算机知识和应用能力等级考试正有效地推动这一目标的实现。1993年12月国家教委考试中心颁布的在全国进行计算机应用能力认证考试文件,这必将进一步推动全社会学习计算机、使用计算机的热潮。与此有关的教材和参考资料的需求与日俱增。

到目前为止,各省、自治区、直辖市都举办了计算机等级考试,之类的书为数不少,本书的出版正是在充分吸收先期出版的同类图书的优点、克服存在的弊病的基础上,推陈出新、更上一层楼。

为了达到这个目的,我们在编撰过程中特别注意了以下三个问题。

(1)适当放低起点,但不降低总的要求,充分贯彻《国家教育委员会全国计算机等级考试大纲》的要求,兼顾目前存在的多种等级考试的要求,循序渐进,深入浅出,对基本内容讲透、讲够,对于易出的错误,给予明确指出,同时对《大纲》进行适当扩充,以保证该书具有一定的实用性与超前性。

(2)注意考试科目的基本知识讲授,在掌握基础知识的同时,适当地进行基本技能的训练,而不以“习题”和“模拟试题”为主。我们认为扎扎实实的掌握基本技能,完全可以达到有关考试的要求。

(3)本书适合于读者自学,也适合于有关专业进行课堂教学,每道习题均有答案,习题本着精辟、典型的原则进行收录。

(4)本书本着实用、广谱的原则,兼容各级各类考试的要求,适合于以下几种考试:

- 各省、市、自治区组织的非计算机专业计算机等级考试
- 国家教委考试中心的非计算机专业计算机能力考试
- 国家教委考试中心的计算机专业的计算机水平和资格考试

全国非计算机专业等级考试、自学考试辅导丛书编委会

本书阅读方法

◁必学/了解

这部分内容是基础性的和稍具理论性的,了解它们是为了学习后续内容的需要。

◁必学/重点

这部分内容是实质性的,要求准确地理解,熟练地掌握,读者阅读时可多加推敲。

◁自学答疑

这部分的内容收集了初学者易产生的一些概念错误、操作错误,它是本书的画龙点睛之处,对于缩短学习周期,提高学习效率有重要指导意义,应认真对待。

◁选 学

这部分的内容超出了考试大纲的要求,若读者在实际应用中遇到这些问题或有空余的时间,不妨一看。

◁编程技巧

这部分是一些极具实用性的技巧,当你“山穷水尽疑无路”时,它也许能让你有“柳暗花明又一村”的感觉。

◁作 业

这部分收集了一些典型的习题,要求认真完成。

◁答 案

本部分是“作业”部分的答案,也许你的答案和书中的答案不一样,这很正常,也许你的答案更先进(向你祝贺),也许我们的答案有误(欢迎指正)。

目 录

第一章 程序设计基础提要	(1)
§ 1.1 程序设计常识	(1)
§ 1.2 程序设计的工具	(3)
§ 1.3 认识 PASCAL 语言.....	(6)
§ 1.4 PASCAL 程序基本语法单位	(9)
§ 1.5 PASCAL 语言上机实践	(13)
第二章 表达式与赋值语句	(16)
§ 2.1 常量与常量.....	(16)
§ 2.2 简单数据类型.....	(20)
§ 2.3 表达式与赋值语句.....	(30)
第三章 输入输出控制	(37)
§ 3.1 标准输入输出文件.....	(37)
§ 3.2 输入控制.....	(38)
§ 3.3 输出控制.....	(40)
§ 3.4 应用举例.....	(45)
第四章 流程控制之一——选择与分支	(49)
§ 4.1 IF 语句	(49)
§ 4.2 CASE 语句	(62)
§ 4.3 GOTO 语句	(69)
§ 4.4 综合应用.....	(71)
第五章 流程控制之二——循环结构	(75)
§ 5.1 WHILE 语句	(75)
§ 5.2 REPEAT 语句	(79)
§ 5.3 FOR 语句	(83)
§ 5.4 循环语句学习答疑.....	(90)
§ 5.5 应用实例选辑.....	(93)
第六章 结构化程序设计及实现——函数与过程	(101)
§ 6.1 函数的初步印象	(101)
§ 6.2 过程的初步印象	(110)
§ 6.3 函数与过程学习辅导	(118)
§ 6.4 变量的作用域	(121)
§ 6.5 子程序嵌套	(123)

§ 6.6	递归调用	(128)
§ 6.7	程序的结构化设计	(133)
第七章	用户自定义类型	(138)
§ 7.1	类型定义	(138)
§ 7.2	枚举类型	(140)
§ 7.3	子界类型	(144)
第八章	数组与字符串	(150)
§ 8.1	一维数组	(150)
§ 8.2	二维数组	(158)
§ 8.3	紧缩数组	(167)
§ 8.4	字符串变量	(169)
§ 8.5	查找与排序	(177)
第九章	PASCAL 特有的类型——集合	(184)
§ 9.1	集合类型	(184)
§ 9.2	集合的运算	(188)
§ 9.3	集合的赋值与输出	(197)
第十章	重构数据结构的工具——记录类型	(201)
§ 10.1	记录类型	(201)
§ 10.2	记录数组及应用	(211)
§ 10.3	变体记录	(217)
第十一章	指针类型	(220)
§ 11.1	初步印象	(220)
§ 11.2	指针的赋值与运算	(224)
§ 11.3	New 过程与动态变量	(225)
§ 11.4	动态数据结构与链表	(229)
§ 11.5	链表节点的删除与插入	(236)
第十二章	文件系统	(242)
§ 12.1	文件的基本概念	(242)
§ 12.2	顺序文件	(246)
§ 12.3	文本文件	(250)
§ 12.4	Turbo PASCAL 的文件操作	(253)

第一章 程序设计基础提要

【导读提要】本章简要地介绍了计算机程序设计的一般常识、高级语言的概念、算法的概念、PASCAL 语言的概貌。§ 1.1~§ 1.3 的内容,对于学过其它高级语言的读者可以略过。

本章中要求熟练掌握的内容有:

- 什么是程序
- 什么是计算机语言
- 什么是算法
- 什么是流程图
- 流程图绘制的方法
- PASCAL 语言的主要语法要素
- 标识符的取名规则
- PASCAL 语言的上机环境

本章中要求了解的内容有:

- PASCAL 语言的执行过程
- 解释与编译的概念

§ 1.1 程序设计常识

一、什么是程序

◁必学/了解

电子计算机能够进行高速地运算,但是,这些运行必须在人工控制下完成,无法完全自动地进行工作。只有当操作者向计算机输入一定的信息(这种信息必须是计算机能够接受的),它才能按照操作者的要求进行工作,并且得出所需的结果。

那么,人们是怎样通过输入的信息来让计算机工作呢?这就涉及到计算机的工作机制。目前的计算机的工作机制基本上是这样的:

(1) 将需要计算机完成的任务编成一条一条的指令,输入计算机,存放在计算机的存储器中;

(2) 计算机在工作时逐条地从内存中取出指令,然后再执行。

可见计算机是通过一条一条的指令来完成一定的工作的。因此,人们要控制计算机也是通过一条一条的指令来进行的。用计算机术语来说,指令的序列被称为程序,计算机就在人们编制的程序控制下工作。程序是人们意志的体现,它表明了要计算机“做什么”和“怎么做”。同时,程序又是计算机处理问题的灵魂,只有当操作者向计算机输入一定的程序,计算机才能按照程序中规定的步骤工作。失去了程序的控制,计算机便无法发挥其作

用，变成一堆废铜烂铁。

二、什么是计算机语言

◁必学/了解

程序是计算机指令的集合。我们做一件事情时，总要按照一定的步骤一步一步地进行。计算机也是一样的，执行一个程序时总是一条指令一条指令地执行。例如，要求三个数 x 、 y 、 z 的平均数 $average$ ，就可以按照以下的“指令”来进行：

- ① 输入 x ；
- ② 输入 y ；
- ③ 输入 z ；
- ④ 求和 $sum = x + y + z$ ；
- ⑤ 求平均数 $average = sum / 3$ ；
- ⑥ 打印 $average$ 的值。

那么，怎样把以上①~⑥所确定的步骤以一种计算机能够接受的形式输入计算机呢？这就必须克服以下问题：

首先，计算机并不能识别汉语、英语、法语等自然语言。你若强行把以上①~⑥以某种自然语言形式输入计算机，计算机将无法执行你的“程序”。

其次，自然语言的意义往往和具体的语言环境有关。一个句子、一个词语在不同的场合的含义往往不同，具有“二义性”，而计算机不能容忍这种“二义性”。因为它根本不具备智能特性，只能机械地、古板地按照你的规定步骤去做，如果你的程序中存在“二义性”，计算机将不知所措。

正是因为以上的原因，我们期待着一种能够准确无误地表达我们编制的程序含义的、一种能够被计算机和人们所接受的、一种相当严谨、不具有二义性的表达方法的出现。这种表达方法正是我们所说的“计算机语言”。

“计算机语言”是人与计算机之间进行通信的工具，是一种计算机能够接受的信息，它由一些简单的单词符号、数字和严谨的语法组成，能够准确无误地表达程序的含义。它是专门用于人们与计算机之间信息交流的一种特殊语言，在人与计算机之间建立起了一座信息的桥梁。

目前计算机语言的种类很多。总的来说，可以分成机器语言、汇编语言和高级语言三大类。本书要详细介绍的 PASCAL 语言就是一种高级语言。

三、如何设计程序

◁必学/重点

人们通过程序让计算机工作起来，以便处理各种信息，解决各种问题，让计算机为人类服务，所以“程序”是人们意志的体现；反过来说，要让计算机按自己的“意志”进行运算，就必须编制程序。编制程序是本书的核心内容，下面简要地说明一下编制一个程序的过程。

(1) 分析问题。这是编程序的第一步，因为任何程序都是为了解决实际问题而编制的，编程序时不能无的放矢，而要认真考察实际问题，找出解决问题的大致思路。

(2) 提出算法。把第一步中的解决问题的思路进一步明确化、详细化，建立解题需要

的数学或物理模型。也就是把解题步骤一步一步详细地写出来，为下一步用计算机语言来表达这些方法奠定基础。

(3) 编写程序。根据第(2)步的方案用一定的计算机语言把程序写出来。

(4) 上机调试。对编好的程序进行实际检验，发现其中的错误之处，不断加以改正，直到程序能达到预期目的。

§ 1.2 程序设计的工具

一、算 法

◁必学/重点

算法就是对问题求解的方法的精确描述。在进行程序设计时，最关键的问题是算法的提出。因为它直接关系到你写出来的程序的正确性、可靠性。如果没有认真地研究实际的问题，就草率地提出一些不成熟的算法，那么编写出来的程序就可能出现错误或疏忽。

算法作为对解题步骤的精确描述，应具备如下性质。

1. 有穷性

一个算法必须在有限步骤之后结束，而不能无限制地进行下去。因此，在算法中必须给出一个算法结束的条件。

2. 明确性

一个算法中的任何步骤都必须意义明确，不能模棱两可、含混不清。

3. 可执行性

所采用的算法必须能够在计算机上执行，因此，在算法中所有的运算必须是计算机能够执行的基本运算。

4. 有一定的输入与输出

要计算机解决问题时，总是需要输入一些原始的数据；计算机向用户报告结果时，总是要输出一些信息。因此，一个算法中必须有一定的输入与输出。

如何理解算法的特性

◁自学答疑

(1) 有穷性。我们不能指望计算机算出圆周率的精确值，因为 π 是一个无穷不循环小数，无法求出它的精确值。

(2) 明确性。即不允许有二意义性，我们不能在计算机中使用诸如“老张对老王说他的儿子考上了清华大学”这样的有歧义的表达方式，到底是老张的儿子上了大学还是老王的儿子上了大学，无法确知。

(3) 可执行性。我们所要计算机执行的算法，计算机应该能够实现，我们不能提出象“让计算机去煮饭，煮完饭之后再炒菜”之类的算法，它在目前无法实现。

二、流程图

◀必学/重点

流程图是一种比较形象地描述算法的形式。它主要由以下几种框图构成。

(1) 起止框。它是一个两边是弧形的长条形图框。一般出现在流程图的开头和最后，代表算法的开始以及结束，如图 1-1(a)。

(2) 一般处理框。它是一个长方形图框，代表算法中的一般性操作，例如赋值、加、减、乘、除等一般操作，如图 1-1(b)。

(3) 判断框。它是一个菱形图框，代表算法中出现的判断，按照是否满足一定的条件而决定是执行流程线的哪一条，如图 1-1(c)。

(4) 输入输出框。它是一个平行四边形图框，用来表示输入及输出等有关操作，如图 1-1(d)。

(5) 流程线。它是一段带有箭头的直线或者折线，用来将以上几种流程框图连接起来，构成完整程序流程表示，其箭头所指的方向就是流程方向，如图 1-1(e)。

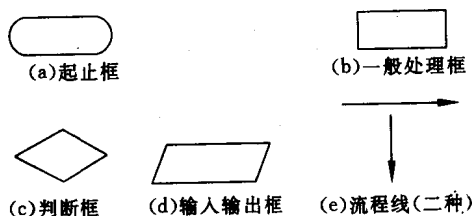


图 1-1

三、流程图表示算法的实例

◀必学/重点

【例 1.2-1】在键盘上输入一个数 num，如果 $0 \leq \text{num} \leq 10$ ，那么就在显示器上显示 num 的值。

图 1-2 就是例 1.2-1 的流程图，先是起止框表示程序开始，之后是输入输出框，表示执行输入数 num 的值的操作。接着流程线指向判断框，判断 num 的值是否大于等于零，再判断 num 的值是否小于等于 10。判断框有两个出口，一个为“是”出口，当判断框中的条件成立时，流程沿着这条流程线执行；否则，如果条件不成立，流程沿“否”流程线执行，若条件成立，则执行输出框中显示 num 的操作，再结束程序。

【例 1.2-2】按照输入的半径值，计算一个圆的面积。

这个问题的算法，可以用流程图 1-3 来描述。

程序开始后，先将 π 的值 3.1416 赋值给变量 Pi，之后输入半径 R 的值。接着，按照求圆面积的公式计算出 Pi 与 R 平方的乘积，并将结果存入 S，于是，S 的值就是该圆的面积。最后，输出圆的面积 S 的值，并结束程序。

有人喜欢用英文来注明流程图，这样将使其和计算机语言更加接近。例如图 1-3 也可以画为图 1-4。

其中，BEGIN 是“开始”的意思；Read 是“输入”的意思；Write 表示“打印输出”；END 表示“结束”。

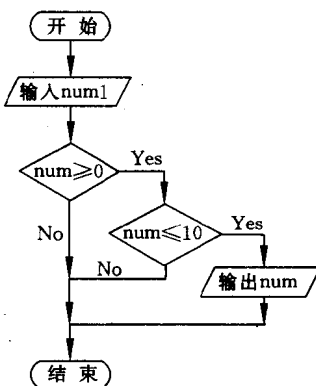


图 1-2

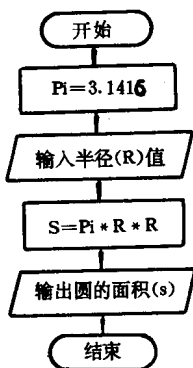


图 1-3

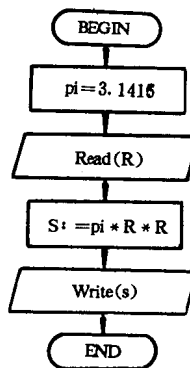


图 1-4

注意事项

◁ 自学答疑

用流程图来表示算法的好处在于：

- ① 首先是用图形来表示流程，形象直观，便于阅读，各步操作清楚了；
- ② 其次，它不会产生“歧义性”，即不会有第二种理解方式，造成程序的错误。但是，流程图也不是万能的，它也有如下四条缺点：

- ① 使用流程图缺乏有关数据的表示法；
- ② 使用流程图难以进行逻辑验证；
- ③ 因为流程图的流程线可以任意转向，稍微复杂的程序往往会使用户很难弄清流程的思路；
- ④ 使用流程图不易对算法进行修改以及维护。

程序与算法有什么区别

◁ 自学答疑

程序是可以在计算机上实现的指令序列，它可以直接在计算机上执行。

算法是对解题方法的精确描述，但是它与“程序”还有一定差距。算法体现了一种思

想方法，而不是实现方法。它可以独立于具体的程序或语言而存在，例如：我们有了对某一个问题的算法，可以用 PASCAL 语言来实现，也可以用其它的语言，比如 C、FORTRAN 等语言来实现。

作 业

◁ 必 做

1. 比较两个数 x 和 y 的大小，并且将较大的数打印输出，如果这个程序用流程图来表示，下面哪一个才是正确的（见图 1-5）？

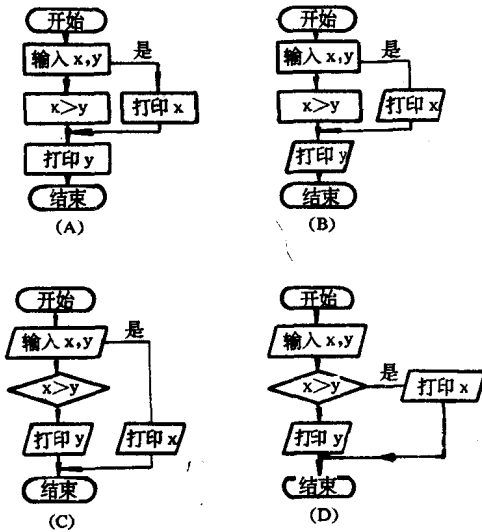


图 1-5 第 1 小题图

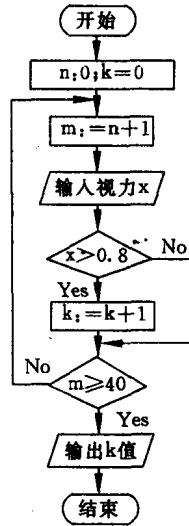


图 1-6

2. 某医院对一个班级中的 40 名学生的视力状况进行抽查，并且要统计出其中视力在 0.8 以下学生的人数，试画出流程图。

答 案

◁ 仅供参考

1. (D) 2. 流程图如图 1-6。

§ 1.3 认识 PASCAL 语言

一、PASCAL 语言概貌

◁ 必学/了解

PASCAL 语言是主要的几种计算机高级语言之一。PASCAL 语言是 60 年代末 70 年代初由瑞士苏黎士大学的 N. Wirth 教授提出的，它是在 ALGOL 60 的基础上发展的一种结构程序设计语言。因为它强大的功能和简明易读的表达方式，所以不少高等学校都选择它做为计算机高级语言教学的第一语言。

PASCAL 语言的第一个显著特点是其简明性和结构化。语言本身具有构成多种控制结构的功能，但表达方式又十分简明严谨。因此，用它编制的程序其结构很清晰，便于验证

其正确性，这特别对一些大型程序有好处。经验证明，要从理论上证明一个程序的正确性往往是困难的。因此，如何设法降低程序的复杂性，提高清晰度，以期得到一个结构良好、易于理解和验证的程序就成为人们十分关注的问题。可以说，PASCAL 是第一个支持这种努力的高级语言。

PASCAL 语言的另一个特点是提供了处理十分广泛的各类数据的强大功能，它不单可直接处理象实数、整数、字符和布尔量等一些简单类型的数据，而且给出一些由这些简单类型构造出复合类型数据（如数组、记录、集合等）的方法。这就大大增强了处理各类应用问题的能力，不单可用于一些传统的数值计算问题，也便于应用于象检索、表格处理、判断、决策等非数值计算的问题。

二、示范程序 Class

◁必学/了解

程序 Class 的功能是输入一个班级 40 名同学的考试成绩，并统计输出其中考分及格学生的人数。程序以 PROGRAM 开始，以句点“.”结束。其中共分为三个部分：程序首部、说明部分和语句部分。

<pre>PROGRAM Class (Input, Output); CONST student=40; pass=60; VAR i, number: Integer; achievement: Real; BEGIN number:=0; FOR i:=1 TO student DO BEGIN WriteLn ('please Input achievement:'); Read (achievement); IF achievement>=pass THEN number:=number+1; END; Write ('The student accessed is', number); END. END.</pre>	<pre>程序首部 } 常量说明 } 说明部分 } 变量说明 } } 语句部分</pre>
---	---

程序中有关部分说明如下：

1. 程序首部

程序首部以固定单词 PROGRAM 开头，称之为保留字。后面 class 称为标识符，它是程序设计者给程序所起的名称。每一个程序都需要有一个名称，程序名称可以是任意的标识符，但一般都是按照该程序的功能来选择适当的名称。程序名后面的圆括号部分称为程序的参数表，它列举程序用到的文件。这里的 Input 和 Output 是两个标准文件，分别代表输入设备（一般指键盘）和输出设备（一般指终端显示屏）。因此，本例中的参数表就表示程序的输入数据是从标准文件 Input 读入，计算的结果输出到标准文件 Output 中去。

在我们进一步学习有关文件的概念之前，如果程序中需要有输入和输出，只须记住在参数表中写上 Input 和 Output 即可（中间用逗号隔开）。当然，如果不需要输入或输出数据，那么参数 Input 或 Output 就可省去不写。这时程序名后面是分号“;”。

2. 说明部分

说明部分是用来对分程序中用到的标识符进行说明和定义。它可以包括标号说明、常量定义、类型定义、变量说明、过程和函数说明五个段，但每个程序并不都需要这五个段，只有需要时才依据具体情况进行某些说明，甚至可以没有说明部分，对于标准 PASCAL 而言，这些段出现的顺序是不能随意颠倒的。如本例就只出现常量定义（以 CONST 开头）和变量说明（以 VAR 开头）两段。其中，常量定义部分定义了两个常量 student 和 pass, student 代表常数 40, pass 代表常数 60。变量说明部分说明了三个变量 i、number 和 achievement, i 和 number 被说明成整数类型变量，achievement 被说明成实数类型变量。

3. 语句部分

语句部分就是程序的可执行部分，它规定完成此程序功能的一系列操作。语句部分必须以 BEGIN 开始，END 结束，我们称之为一对语句括号。因此，语句部分可以说就是用语句括号 BEGIN 和 END 括起来的一系列用分号隔开的语句。其中语句的数量没有限制，完全根据需要而设定。分号是语句之间的分隔符，不是语句的组成部分，一行内可以写多条语句，也可只写一条语句，但语句之间必须有分隔符“;”，最后一个语句与 END 之间可以不要加分号（Turbo PASCAL 中，每个语句之间都必需有分号）。例如，本例中就出现了六条语句，第一条语句 number := 0 是将数 0 赋值给变量 number，表示开始及格的人数是 0 个。后面为一个循环语句，它依次统计 1 到 40 个学生的成绩。第三、四两条语句是要求用户输入某个学生的成绩。第五条语句是用来判断输入的成绩是否及格，如果及格则及格人数 number 的值增加 1。最后一条语句就输出及格的人数，即 number 的值。

三、程序的书写格式

◁必学/重点

这里还有必要对程序的书写格式作些说明。诚然，程序的书写格式一般不影响程序的功能，但为了使程序的结构清晰，易于阅读，注意适当的书写格式是必要的。我们在本书中都采用紧缩对齐格式书写，而且适当加插一些空格，并用大写字母拼写保留字以便与其它标识符有所区别。语句括号 BEGIN 和 END 应成对对齐，里面的语句又需再向右缩进，使得层次分明，便于阅读。希望读者都能参照这种格式书写自己的程序。

作 业

◁必 做

() 一个完整的 PASCAL 程序一般是由以下哪些部分构成的？

- (A) 程序首部、变量说明部分和语句部分
- (B) 标号说明部分、说明部分和语句部分
- (C) 程序首部、说明部分和类型定义部分
- (D) 程序首部、说明部分和语句部分

答 案

◁仅供参考

(D)

§ 1.4 PASCAL 语言基本语法单位

所谓基本语法单位，就是具有一定语法意义的，可以按照一定的语法规则进一步构成其它语法单位、语句和程序的基本字符串。

一、PASCAL 语言字符集

◁必学/重点

在介绍基本语法单位之前，首先介绍一下 PASCAL 语言的字符集。某种语言所采用的字符集合称为字符集，它包括常用字符和一些其它符号，见表 1-1。

表 1-1 PASCAL 基本字符集

类别名称	具体符号	个数
英文大写字母	A, B, C, ..., Z	26
英文小写字母	a, b, c, ..., z	26
数字字符	0, 1, 2, 3, 4, ..., 9	10
运算符号	+, -, *, /, ^	5
标点符号	, ; : ' .	5
括 号	() [] { }	6
关系运算	>, <, =	3
其 它	↑ (指针符号) (空格)	2

其中英文字母可以是大写字母也可以是小写字母，它们的意义相同。

这就是 PASCAL 语言的字符集。PASCAL 程序的所有基本语法单位都是由这些字符构成的，如果在一个程序中出现了非上述字符集的字符，则程序将会出错。

由以上字符构成的基本语法单位包括保留字、标识符、数字和特殊符号，以及指示字。

二、保留字

◁必学/重点

保留字是 PASCAL 语言规定的用来构成语句和程序，或者作为特殊操作符的单词。例如，在上一节所举的程序例子 Class 中，单词 PROGRAM、CONST、VAR、IF、BEGIN 和 END 等都是保留字。标准 PASCAL 语言中基本的保留字共有 35 个，见表 1-2。之所以称之为保留字，是因为这些单词含义特定，系统专用，不允许用户在程序中给这些单词赋予其它的含义。除了这些标准 PASCAL 的保留字以外，某些不同版本的 PASCAL 语言还增加了一些其它的保留字来扩充标准 PASCAL 的某些功能。关于这些保留字的具体含义，将在以后的章节中一一介绍。