



21世纪高等学校计算机学科系列教材

Pascal语言 程序设计

张长海 编著

全国高等学校计算机教育研究会
课程与教材建设委员会推荐出版



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

URL: <http://www.phei.com.cn>

21 世纪高等学校计算机学科系列教材

Pascal 语言程序设计

张长海 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书全面系统地介绍了国际标准 ISO—7185 定义的 Pascal 语言及其程序设计方法;用 BNF 表示 Pascal 的语法;引进 PAD 表示程序逻辑。全书共分 13 章,主要内容包括:BNF、PAD、Pascal 的各种词法单位、数据类型、语句、过程和函数、作用域及其规则,程序开发和结构化程序设计。每章都包含大量例题,并附有大量习题,以利于读者掌握语言概念、提高编程能力。

本书整体结构良好,图文并茂,知识体系新颖完整,概念准确;注重对读者进行程序设计方法及算法的训练,力求体现“结构化程序设计”思想,注意培养和训练学生的程序设计风格。

本书可作为高等院校计算机系各专业“高级语言程序设计”课程的教材和参考书,也可以供其他专业学生以及从事计算机工作的有关人员阅读参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,翻版必究。

图书在版编目(CIP)数据

Pascal 语言程序设计/张长海编著. - 北京:电子工业出版社,2001.1

21 世纪高等学校计算机学科系列教材

ISBN 7-5053-6371-9

I .P… II . 张… III .Pascal 语言-程序设计-高等学校-教材 IV .TP312

中国版本图书馆 CIP 数据核字(2000)第 74881 号

丛 书 名: 21 世纪高等学校计算机学科系列教材

书 名: Pascal 语言程序设计

编 著 者: 张长海

责任编辑: 张荣琴 束传政

排版制作: 电子工业出版社计算机排版室监制

印 刷 者: 北京东光印刷厂

出版发行: 电子工业出版社 URL: <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1 092 1/16 印张: 21.75 字数: 550.4 千字

版 次: 2001 年 1 月第 1 版 2001 年 1 月第 1 次印刷

书 号: ISBN 7-5053-6371-9
TP·3457

印 数: 6 000 册 定价: 27.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换;
若书店售缺,请与本社发行部联系调换。电话 68279077

序

为什么要组织《21 世纪高等学校计算机学科系列教材》? 根据什么组织这套教材? 这些都是在这篇序言中要回答的问题。

计算机学科是一个飞速发展的学科,尤其是近十年来,计算机向高度集成化、网络化和多媒体化发展的速度一日千里。但是,从另一方面来看,目前高等学校的计算机教育,特别是教材建设,远远落后于现实的需要。现在的教材主要是根据《计算机学科教学计划 1993》的要求组织编写的。这个教学计划,在制订过程中主要参照了美国 IEEE 和 ACM《Computing Curricula 1991》(简称 CC 1991)。

10 年来,计算机学科已有了长足发展,这就要求高等学校计算机教育必须跟上形势发展的需要,在课程建设和教材建设上作出相应调整,以适应 21 世纪高等学校计算机教育的要求。这是组织这套教材的初衷。

为了组织好这套教材,全国高等学校计算机教育研究会课程与教材建设委员会在天津召开了“全国高等学校计算机学科课程与教材建设研讨会”;在北京召开了“教材编写大纲研讨会”。在这两次会议上,代表们深入地研讨了全国高校计算机专业教学指导委员会、中国计算机学会教育委员会制订的《计算机学科教学计划 2000》和美国 IEEE、ACM《Computing Curricula 2001》(简称 CC 2001)。这是这套教材参照的主要依据。

IEEE 和 ACM 发表的 CC 2001 是在总结了从 CC 1991 到现在,计算机学科十年来发展的主要成果的基础上诞生的。它认为 21 世纪计算机学科应包括 14 个主科目,其中 12 个主科目为核心主科目。它们是:算法与分析(AL)、体系结构(AR)、离散结构(DS)、计算科学(CN)、图形学、可视化、多媒体(GR)、网络计算(NC)、人机交互(HC)、信息管理(IM)、智能系统(IS)、操作系统(OS)、程序设计基础(PF)、程序设计语言(PL)、软件工程(SE)、社会、道德、法律和专业问题(SP)。其中除 CN 和 GR 为非核心主科目外,其他 12 项均为核心主科目。

将 CC 2001 与 CC 1991 比较可看出:

在 CC 1991 中,离散结构只作为数学基础提出,而在 CC 2001 中,则作为核心主科目提出,显然,提高了它在计算机学科中的地位。

在 CC 1991 中,未提及网络计算,而在 CC 2001 中,则作为核心主科目提出,以适应网络技术飞速发展的需求。

图形学、可视化与多媒体也是为适应发展要求新增加的内容。

除此之外,CC 2001 在下述 5 个方面作了调整:

程序设计语言引论调整为程序设计基础;将人-机通信调整为人机交互;将人工智能与机器人学调整为智能系统;将数据库与信息检索调整为信息管理;将数值与符号计算调整为计算科学。

显然,这些变化使 CC 2001 更具有科学性,也更好地适应了学科发展的需要。

在组织这套教材的过程中,充分考虑了这些变化和调整,在软件和硬件的课程体系、界面划分时均做了相应的调整,使整套教材更具有科学性和实用性。

另外,还要说明一点,教材建设既要满足必修课的要求,又要满足限选课和任选课的要求。

因此,这套教材按系列组织,反映了整个计算机学科的要求;采用大拼盘结构,对于美国 IEEE 和 ACM 发表的 CC 2001 中提出的 14 个主科目,这套系列教材均涵盖,能够满足不同层次院校、不同教学计划的要求,各校可根据自己的需求进行选拼使用。

这套教材由全国高等学校计算机教育研究会课程与教材建设委员会主任李大友教授精心策划和组织,参加这套教材组织工作的还有哈尔滨工业大学蒋宗礼教授、西安交通大学董渭清副教授、武汉大学张焕国教授、吉林大学张长海教授、福州大学王晓东教授、太原理工大学余雪丽教授、西安电子科技大学王闵教授等。

本套教材的编者均为具有丰富教学实践经验的专家和教授。所编教材体系结构严谨、层次清晰、概念准确、论理充分、理论联系实际、深入浅出、通俗易懂。

李大友

2000 年 6 月

前 言

本书应用于“高级语言程序设计”课。“高级语言程序设计”是计算机系的专业基础课,在计算机专业教学中占有重要地位。学好该课既可以为后续课程的学习打下良好的基础,又可以对以后学生的程序设计技术、技巧、风格和习惯产生深远影响。

本书力图在知识的深度、广度和知识结构上作出合理的安排,试图使学生既具有较强的编程能力,又能掌握高级语言 Pascal 本身严格的语法和语义;同时在知识结构、知识面上尽量做到广泛、全面。

本书作者从事计算机教学已经 20 余年,开过 10 多门计算机方面的课程。尤其近年来专门对 Pascal 语言进行了深入研究,仔细研究了从国际标准 ISO - 7185 到国家标准 GB7591 - 87 等有关 Pascal 语言的各种标准文本。曾 10 余次为吉林大学计算机科学系本科生主讲“Pascal 程序设计”课。本书是作者十几年教学实践的总结,是作者在其教学笔记和自编教材基础上整理编辑产生的。作为吉林大学计算机系本科生教材已经使用多年,不论教学效果、学生反映,还是社会反映都很好。

作为大学本科计算机专业基础课教材,本书具有如下特点:

1. 全书整体结构好,知识体系新颖完整,章节安排合理,并注意由浅入深地介绍 Pascal 语言的概念和程序设计知识(例如过程的概念分 3 章逐步介绍)。

2. 注重对学生进行严格的抽象思维训练,严格按照国际标准 ISO - 7185 介绍 Pascal 语言,并使用 BNF 表示语法,使用自然语言叙述语义,从而使得对 Pascal 语言的语法、语义的描述严格、细致、准确,并且形式化,为后续课程(例如编译原理)打下良好基础。

3. 注重对学生进行程序设计方法及算法的训练,力图做到严格的理论与具体方法及算法有机结合。全书配备大量例题,一方面,在讲解例题时着重于算法的构造,以便训练学生的编程能力;另一方面,概念的介绍都以例题作引导,从具体实例出发,使得概念引进自然,容易理解。另外,本书还配备大量的习题,以便学生课后练习和进一步提高用。

4. 全书自始至终贯穿结构化程序设计思想,所有例题都具有良好的结构和程序设计风格。目的是给学生一个示范,以便使学生从开始学习程序设计就养成一个良好的程序设计习惯和风格。

5. 本书图文并茂,引进 PAD 表示程序逻辑。PAD 的结构比传统的流程图、NS 图等都好,同时也比直接用程序表示算法更直观,易于理解。

全书共分 13 章。第 1 章介绍计算机基本知识,引进 BNF 和 PAD;第 2 章介绍 Pascal 的基本符号、单词、数据及其类型;第 3 章介绍常量、变量、赋值语句、I/O、及类型一致性;第 4 章引进过程和函数的概念;第 5 章讲述流程控制;第 6 章开始接触结构型数据类型并讲述数组;第 7 章进一步介绍子程序概念,讲述值参数和变量参数的区别、作用域概念和规则、以及递归程序设计;第 8 章讲集合;第 9 章介绍程序开发和结构化程序设计,包括结构化程序设计原则、程序风格、自顶向下逐步求精的程序设计技术、程序正确性、可移植性、文档,最后介绍穷举法和试探法;第 10 章介绍文件;第 11 章介绍记录;第 12 章介绍指针和动态数据结构;第 13 章讲述几个较深入的问题,包括一致数组参数、过程和函数作参数、函数副作用等。

在本书的编写过程中作者参阅并引用了国内、国际诸多同行的文章、著作,本人在此向他们致意,并恕不能一一列举、标明,在参考文献中只给出了一些;在本书的成书出版过程中得到中国计算机学会教育专业委员会、全国高等学校计算机教育研究会课程与教材建设委员会、电子工业出版社的帮助和大力支持,作者在此向他们一并表示感谢。

限于作者学术水平有限,错误和不足在所难免,敬请各位读者批评指正。本人表示由衷地感谢。

作者

2000 年于长春

目 录

第 1 章 基本知识	1
1.1 计算机的基本结构	1
1.2 程序设计语言	1
1.2.1 机器语言	2
1.2.2 汇编语言	2
1.2.3 高级语言	3
1.2.4 汇编语言和高级语言程序的执行	3
1.3 Pascal 语言简况	3
1.4 程序语言的形式描述	5
1.4.1 法、语义	5
1.4.2 BNF	5
1.4.3 文法的其它表示法	7
1.5 Pascal 程序结构	9
1.6 算法及其描述工具 PAD	10
1.6.1 算法	10
1.6.2 PAD	11
本章小结	13
习题一	13
第 2 章 基本符号、数据及其类型	16
2.1 基本符号	16
2.1.1 字符集	16
2.1.2 特定符号	16
2.1.3 指示字	17
2.1.4 标识符	17
2.1.5 标号	18
2.1.6 数	18
2.1.7 字符串	20
2.1.8 记号分隔符	20
2.2 数据及其类型	21
2.2.1 integer 型	22
2.2.2 real 型	23
2.2.3 char 类型	24
2.2.4 Boolean 类型	24
2.3 类型定义	25
2.4 枚举类型	26

2.5 子界型类型	27
2.6 值的可比性及关系运算	28
本章小结	29
习题二	29
第3章 简单程序	32
3.1 常量及常量定义	32
3.2 变量及变量说明	34
3.3 标准函数	36
3.4 表达式	39
3.4.1 表达式的构成形式	40
3.4.2 表达式值的计算	41
3.4.3 表达式的类型	42
3.5 语句	43
3.6 赋值语句	43
3.7 类型一致性	45
3.8 输入输出	48
3.8.1 标准函数	48
3.8.2 输入	49
3.8.3 输出	51
3.8.4 输出表项	52
本章小结	56
习题三	56
第4章 过程与函数	61
4.1 带子程序的 Pascal 程序	61
4.2 函数	64
4.2.1 调用函数——函数命名符	64
4.2.2 定义函数——函数说明	64
4.3 过程	67
4.3.1 调用过程——过程语句	68
4.3.2 定义过程——过程说明	69
4.4 程序设计实例	70
本章小结	74
习题四	74
第5章 流程控制	77
5.1 顺序结构——复合语句	77
5.2 分支结构——分支程序设计	77
5.2.1 IF 语句	77
5.2.2 CASE 语句	81
5.3 重复结构——循环程序设计	83
5.3.1 WHILE 语句	84

5.3.2	UNTIL 语句	85
5.3.3	FOR 语句	88
5.4	程序设计实例	92
	本章小结	99
	习题五	99
第 6 章	数组	108
6.1	结构数据类型	108
6.2	数组类型	108
6.2.1	数组类型定义	108
6.2.2	数组变量与下标变量	109
6.2.3	有关运算问题	110
6.2.4	应注意的问题	112
6.3	多维数组	113
6.4	程序设计实例——数组在程序设计中的应用	115
6.5	紧缩数组	131
6.6	字符串	133
	本章小结	134
	习题六	134
第 7 章	再论过程——参数、作用域和递归	144
7.1	值参数与变量参数	144
7.2	作用域	149
7.2.1	程序的层次结构	149
7.2.2	定义点及定义点区域	150
7.2.3	作用域及作用域规则	151
7.2.4	定义性出现和使用性出现	153
7.2.5	局部量和全局量	153
7.2.6	作用域例题	153
7.3	无参过程与函数	157
7.4	递归	158
7.4.1	递归程序	158
7.4.2	递归程序设计	159
7.4.3	间接递归	164
7.4.4	递归程序执行过程	170
	本章小结	180
	习题七	180
第 8 章	集合	187
8.1	集合类型	187
8.2	集合值的表示——集合构造器	187
8.3	集合运算	188
8.4	集合应用及例题	190

本章小结	192
习题八	192
第 9 章 程序开发和结构化程序设计	195
9.1 GOTO 和标号	195
9.1.1 带标号的语句	195
9.1.2 GOTO 语句	195
9.1.3 标号说明	196
9.1.4 对标号的认识	196
9.1.5 应注意的问题	197
9.2 空语句	199
9.3 结构化程序设计原则	199
9.4 程序风格	200
9.4.1 良好的行文格式	201
9.4.2 用合适的助忆名来命名标识符	203
9.4.3 注释	203
9.4.4 对程序说明的建议	204
9.5 程序的正确性	204
9.5.1 错误种类	204
9.5.2 程序测试和验证	205
9.5.3 测试方法	206
9.6 可移植性	206
9.7 文档	207
9.8 自顶向下逐步求精的程序设计技术	208
9.8.1 自顶向下、逐步求精	208
9.8.2 求精过程的表示	210
9.9 受限排列组合——穷举法与试探法	218
本章小结	229
习题九	229
第 10 章 文件	236
10.1 文件类型定义及变量说明	236
10.2 文件操作	237
10.2.1 文件状态描述	237
10.2.2 写操作	239
10.2.3 读操作	240
10.2.4 文件操作例题	242
10.3 Pascal 文件变量与操作系统文件的联系	243
10.3.1 程序参数方式	243
10.3.2 assign 方式	244
10.3.3 open 方式	244
10.4 text 文件	245

10.4.1 隐含的数据类型翻译功能	246
10.4.2 text 文件的行式结构	247
10.4.3 text 文件与一般文件区别	249
10.4.4 text 文件例题	249
本章小结	251
习题十	251
第 11 章 记录	255
11.1 记录类型	255
11.2 记录变量及其成分变量	256
11.3 WITH 语句	260
11.4 变体	261
11.4.1 带变体的记录	262
11.4.2 变体部分的域与固定部分的域之间的关系	266
11.4.3 带变体记录结构的动态确定	268
11.4.4 CASE 语句与记录变体	270
11.5 作用域和限制	270
11.6 程序设计实例	272
本章小结	277
习题十一	277
第 12 章 指针及动态数据结构	281
12.1 指针类型	282
12.2 动态数据结构——指针应用	287
12.2.1 栈(stack)	287
12.2.2 队列(queue)	288
12.2.3 链表	289
12.2.4 树	292
12.3 new、dispose 与带变体的记录	297
12.4 动态数据结构的程序设计实例	298
本章小结	311
习题十二	311
第 13 章 三论过程——几个较深入的问题	317
13.1 一致数组参数	317
13.1.1 形式参数与实在参数的对应	319
13.1.2 一致数组参数结合动作	320
13.2 过程和函数作参数	321
13.3 函数副作用	323
13.4 形式参数作实在参数	324
13.5 标识符冲突	324
13.6 参数结合顺序	325
13.7 转出过程体	327

本章小结.....	328
习题十三	328
综合习题	329
附录一 ACSII 字符集	332
附录二 Pascal 运算符表	333
参考文献	334

第1章 基本知识

现代计算机从出现至今不过 50 年左右的时间,但其发展速度是任何一种新技术都不可比拟的,目前计算机已经渗透到各个领域。本书通过介绍一种计算机语言 Pascal,学习怎样用 Pascal 编写计算机程序,以此来向大家揭示如何使用计算机解决科技、生产、事务处理等方面的问题。在具体介绍 Pascal 语言及其程序设计之前,先简略介绍一下有关计算机的基本知识。

1.1 计算机的基本结构

一个计算机系统由中央处理器(Center Processing Unit, CPU),内存储器及若干外部设备组成,如图 1.1 所示。

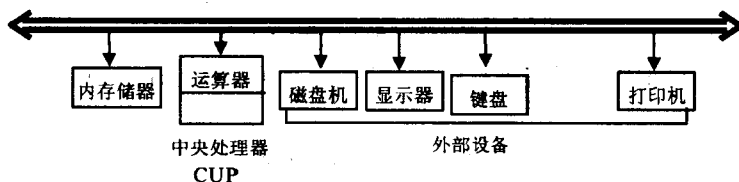


图 1.1 计算机的组成

- 内存储器是计算机的记忆装置,用来存放程序和参与运算的数据。一台计算机的存储器能够存储大量的信息。内存储器分成若干单元,并顺序地给每个单元编以一定的号码,称为该单元的“地址”。按单元地址可以访问某一单元的信息。一般以 8 位二进制为编址单位,也就是说一个单元可以存放 8 位二进制信息,称作一个字节(byte),二个或四个字节称作一个机器字(word)。有时还称一位二进制为一个位(bit)。

- 中央处理器包括控制器和运算器两部分。

控制器控制整个计算机系统工作。它依次从存储器中取出一条条指令,对指令进行分析,然后按指令的要求向计算机的其他部件发出命令,指挥整个计算机系统工作。

运算器是进行算术和逻辑运算的。它接收由控制器发来的运算命令,依据命令的要求进行相应的算术和逻辑运算。

- 外部设备又称计算机的输入输出设备,也简称为 I/O 设备。它们是计算机与外界交流信息的装置。人们通过 I/O 设备将程序、数据等信息送入计算机中。计算机通过 I/O 设备将计算结果、控制信号等信息输出出来,供人们使用。常见的 I/O 设备有终端显示器、终端键盘、鼠标、打印机、磁盘机、磁带机、绘图仪等等。

1.2 程序设计语言

一个庞大的计算机系统是怎样有条不紊地工作的呢? 答案是:计算机系统的工作是由事

先设计好的程序来控制的。人们首先按自己的需要把让计算机做的工作排成计算机程序,并把程序送入计算机,然后启动计算机的执行程序。计算机的控制器从程序的第一条指令开始,顺序地逐条取出指令进行解释,然后按指令的规定和要求指挥整个计算机系统工作,从而完成人们设想的需要计算机完成的工作。

程序是一个指令序列,也就是用指令序列排成的一个工作顺序、工作步骤。我们平常也使用程序这个名词,例如运动会程序等。计算机程序是用计算机指令为计算机排定的工作顺序、工作步骤。

为计算机编排程序的过程称为程序设计。

描述程序必须使用一种语言。程序设计语言是指用于编写计算机程序的语言。一般地,人们将程序设计语言分成三类:机器语言、汇编语言、高级语言。

1.2.1 机器语言

机器语言由能被计算机直接执行的机器指令组成,每条机器指令是一串二进制代码。用机器语言编出的程序是一串二进制代码序列。例如若计算:

$$Y = \begin{cases} X+1 & \text{若 } X < Y \\ X-1 & \text{若 } X \geq Y \end{cases}$$

用 Pentium 机器语言可编出如下程序片段(设程序从 100 号单元开始;X、Y 分别占用 116、118 号单元)。

```
1010 1001 0001 0110 0000 0001
0011 1100 0001 1000 0000 0001
0111 1100 0000 0101
0010 1101 0001 0101 0000 0000
1110 1010 0000 0011
0000 0101 0001 0101 0000 0000
1010 0011 0001 1000 0000 0001
.....
0000 0000 0000 0000
0000 0000 0000 0000
```

用机器语言编程序显然十分困难,编出的程序不但容易出错、调试极为困难,而且程序本身也极不好读。基于上述原因,人们引进了汇编语言。

1.2.2 汇编语言

汇编语言是符号化了的机器语言,也就是引进一些助记符表示机器指令中的操作码,地址等等。下面给出了 1.2.1 节中例题的 Pentium 汇编语言程序片段:

```
MOV AX,X
CMP AX,Y
JLS1
SUB AX,15
JMP S2
S1: ADD AX,15
```

S2: MOV Y, AX

... ..

XDW?

YDW?

汇编语言程序显然比机器语言程序进了一步,它比较好读好懂,写起来也显然比二进制代码程序方便得多了。其原因在于用符号助记符代替了单调的二进制代码。

虽然汇编语言比机器语言前进了一步,但是它仍然十分烦琐,并且仍然依赖于具体的计算机,程序不便于移植。因此人们又进一步引进了高级语言。

1.2.3 高级语言

高级语言程序不依赖于具体计算机,它用比较接近于自然语言或专业语言的方式描述操作。例如使用 Pascal 语言完成 1.2.1 例中同样的计算,可用如下语句:

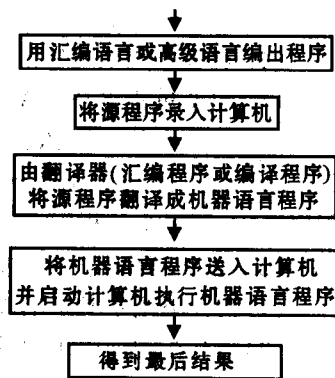
```
IF x < y THEN y: = x + 15
ELSE y: = x - 15
```

显然这种程序十分好读,它几乎就类似于英语句子。这种程序编起来自然很轻松,而且它还具有通用性,可以在不同机器上运行,程序十分便于移植。

1.2.4 汇编语言和高级语言程序的执行

尽管汇编语言和高级语言编写程序比机器语言方便得多,但是不幸的是计算机只认识二进制代码(机器语言),既不懂汇编语言的符号代码,也不懂高级语言的近似自然语言的语句。为了使计算机懂得汇编语言及高级语言程序,人们设计了翻译器。由翻译器把用汇编语言或高级语言写的程序(称源程序)翻译成等价的机器语言程序(称目标程序)。翻译器也是一个程序,并且称汇编语言的翻译器为汇编程序,高级语言的翻译器为编译程序。

如图 1.2 所示,用汇编语言或高级语言编程序解题的过程是:首先用汇编语言或高级语言编出程序,然后把程序录入计算机,接着启动翻译器,由翻译器将相应的汇编语言程序或高级语言程序翻译成机器语言程序。最后再把机器语言程序送入计算机,并启动计算机执行机器语言程序,完成人们预先要求计算机做的工作。



1.3 Pascal 语言简况

Pascal 语言是瑞士苏黎世联邦工业大学 Niklaus Wirth 教授于 1968 年为讲授程序设计课而设计的,它以法国数学家 Blaise Pascal(曾经发明了世界上第一台机械计算机——“加法机”)的名字命名,最早发表于 1971 年瑞士的《ETH》杂志上。Pascal 是从 ALGOL₆₀ 衍生而来的,但决不是 ALGOL₆₀ 的简单扩充。它是由 E. W. Dijkstra 和 C. A. R. Hoare 定义的结构程序设计概念的系统体现,是它的第一个程序设计语言。从这个意义上讲, Pascal 是程序设计语言发展史上的一个里程碑。

由于 Pascal 是系统地体现结构程序设计概念的第一个语言,所以它一发表便立即引起计

图 1.2 汇编语言、高级语言解题过程

计算机界的广泛重视,很快便从课堂上走出来,进入实用领域,并开始了它的标准化工作。1977年英国标准协会(BSI)成立了一个工作小组,负责 Pascal 的标准化工作。该小组于 1978 年 9 月发表了一个征求意见稿 DPS/13/14,即“工作草稿/3”。1978 年 3 月 BSI 向国际标准化组织(ISO)建议,Pascal 的标准化工作应该纳入 ISO 的工作日程,并被采纳。以后 Pascal 的标准化工作由 ISO 接手。ISO 于 1980 年公布了 Pascal 标准草案并征求意见,于 1983 年公布了最终的 Pascal 标准——ISO—7185。我国也于 1987 年制定了我国的 Pascal 国家标准(GB7591—87)。

关于 Pascal 语言的特征,N. Wirth 教授有一段经典的话,摘录如下:

1. 变量说明是强制性的。
2. 某些基本字(例如 BEGIN, END, REPEAT 等)是“保留”的,不能作标识符。
3. 分号(;)看作是语句的分隔符,而不是语句的终止符(如同 PL/1 语言一样)。
4. 标准数据类型是整数、实数、字符和布尔类型。基本数据构造工具有数组、记录(对应于 COBOL 和 PL/1 语言的“结构”)、集合和文件。这些结构可以组合和嵌套以形成集合的数组、记录的文件等等。数据还可以动态地分配并由指针访问。
5. 集合 SET 数据结构提供了类似于 PL/1 语言的“位串”的能力。
6. 数组可以具有任意维数和任意的界;数组的界是常量,即没有动态数组。
7. 像 FORTRAN、ALGOL 和 PL/1 语言一样,存在转向语句。标号是无符号整数,必须加以说明。
8. 复合语句如同 ALGOL 语言,相应于 PL/1 语言里的 DO 群。
9. ALGOL 语言的开关和 FORTRAN 语言的计算转语句的功能在 Pascal 中由分情况语句表示。
10. 循环语句相当于 FORTRAN 的 DO 循环,它的步长只能是 1(TO)或 -1(DOWN-TO),并仅当控制变量的值落在界内时执行。因此,被控制的语句可能完全没有执行。
11. 没有条件表达式和多重赋值。
12. 过程和函数可以递归调用。
13. 对变量不存在 OWN 属性(像 ALGOL 语言一样)。
14. 参数有值调用和引用调用,没有名字调用。
15. 就不存在匿名的分程序来说,这里的“分程序结构”不同于 ALGOL 和 PL/1 语言的分程序结构,即每个分程序都给一个名字,并且做成一个过程。
16. 常量,变量等所有对象必须在它们被引用之前加以说明。但允许两个例外:
 - (1)指针类型定义里的类型标识符;
 - (2)当存在向前引用时的过程与函数调用。

许多人初次接触 Pascal 语言,往往感叹它缺少某些“中意的特色”,例如方幂运算、串的并列、动态数组、布尔值的算术运算、自动的类型转换和省缺说明等,这些都不是疏忽遗漏,而是有意删除的。在某些场合下,有它们反而会招致程序设计解法的效率不高;在另外一些场合下,又会感到它们不利于达到程序设计“清晰可靠”和“风度优美”的目标。最后,还得考虑对繁多的程序设计功能进行严格挑选,以保证编译程序比较紧凑和高效。