

# PASCAL 语言讲义

## (RT-11)

上海电子计算机厂

## 前 言

在今年三月底至四月初办的嘉定用户学习班上，许多用户曾提出要求，希望我们能提供一本 Pascal 讲义，以便他们学习和使用 Pascal 语言用。为了满足这一要求，我们从现有的各种资料上分别选取有关章节，并加上 DJS-185 机及 PDP-11/23 上 RT-11 Pascal 的扩充与限制而整理、汇编成本讲义。

本讲义的第一至第六章是 Pascal 语言的基本内容，属于我们用户学习班的讨论范围；第七章汇集了二个较大的程序设计实例，其目的是提供用户今后使用 Pascal 语言书写应用软件和编译程序时参考之用；第八章仅对 DJS-185 及 PDP-11/23 所配备的 Pascal 编译系统作一简单介绍，以让用户对它有个初步了解，而更具体的使用方法和有关细节尚须阅读“DJS-180 系列机 Pascal-1 语言使用说明”及 RT-11 操作系统的有关资料。

最后借此机会向北京工业大学计算机科学系的丘玉圃等老师表示谢意，因为他的讲义和译文的部份章节是本讲义的组成部份；也向复旦大学计算机科学系的夏宽理老师表示谢意，他的教材的部份内容也是本讲义的组成部份。

由于时间匆促，水平限制，讲义中难免存在我们的错误。谨请用户予以指正。

上海电子计算机厂      郭玉英 顾奇 董国壮  
于八三、五、一

## 绪 论

### 一、Pascal 语言产生的历史背景

“计算机软件”一词是1964年提出的，这时，从软件发展史看它又正进入一个所谓软件危机时期。系统越来越大，错误越来越多，软件产品交付日期越来越得不到保证。软件给人的印象是不可靠，使人认识到建造大型软件是一项尚未掌握的事业。1968年召开的NATO会议，提出“软件工程”概念问题。中心是如何研究大型软件，提高软件的生产率，使其具有较高的可靠性和可维护性。这涉及程序设计方法论、工具、组织和规模等各方面的问題。

从方法论方面，荷兰计算机科学家，Turing奖获得者E.W. Dijkstra提出了“结构程序设计”思想。这是为了使程序具有合理结构以便保证和验证其正确性而规定的一套进行程序设计的方法。用结构程序设计的方法设计出的程序称为结构化程序。语言成分反映结构程序设计的要求和限制，从而便于书写结构化程序。同时使用它写出的程序易于保证其正确性，这样的一种语言称为结构程序设计语言。Pascal是一种结构程序设计语言。

### 二、Pascal是怎样的一种程序设计语言

在这样的历史背景下，1971年产生了Pascal语言。Pascal语言是N. Wirth教授设计的，以历史上著名数学家B. Pascal的名字命名。

Pascal语言是七十年代最有影响和最重要的一种程序设计语言，它是作为Algol 60的后继发展而来。Algol 60的另一

后继是Algol 68，但它走的却是另一道路。

Algol 68 注重于语言的通用性、灵活性和功能的广泛性，而 Pascal 注重于语言的可靠性和易于验证，概念的清晰性和实现的简化性。

Pascal 是一种结构程序设计语言。

Pascal 又是一种系统程序设计语言，可以用来书写顺序型系统软件，如编译程序等。

Pascal 又是一种自编译语言。这使得它可以采用特殊的方法来实现，从而提高了可靠性。

Pascal 语言比 Fortran、PL/I 是更不依赖于机器的语言，比 Fortran、Algol 60 是功能更强的语言；比 PL/I 或 Algol 60 又是较简单的语言，因为它的编译程序没有必要是庞大的和多遍描述等。

### 三、适用范围

Pascal 除适用于科学计算外，特别适用于

(1) 教学。它使程序设计的基本概念和结构可以用 Pascal 所提供的一种系统的、精确的、而且合理的方法来描述。例如，计算机科学系中的课程，如数据结构、编译技术、离散数学中的概念、算法，可以用 Pascal 来描述。

(2) 书写顺序型系统软件与应用软件。Pascal 语言具备书写顺序型系统软件与应用软件的各种语法成分，具有丰富的数据类型，且提供构造新类型的方法，因此可以描述各种各样的数据结构。

#### 四·现状

Pascal 语言的出现虽只有十年的历史,但它已是目前世界上流行最广泛的一种程序设计语言。从大型计算机到微型计算机,大都配备了Pascal编译程序。在虚拟机上实现的Pascal编译程序已成为商品。

目前,国际标准化组织(ISO)正在对Pascal语言进行标准化工作〔3〕,最近(1981年10月),又提出Pascal语言的修正稿〔4〕。

鉴于Pascal语言的优点,它将会在我国得到推广和应用。

## 目 录

|                                   |      |
|-----------------------------------|------|
| 绪 论 .....                         | 1    |
| 第一章 PASCAL 语言概述 .....             | 1-1  |
| § 1 程序举例 .....                    | 1-1  |
| § 2 程序结构、语言要素和特点 .....            | 1-2  |
| § 3 特殊记号 .....                    | 1-6  |
| § 4 标识符 .....                     | 1-9  |
| § 5 数、字符串、常量定义、注解和分隔符 .....       | 1-10 |
| § 6 标准类型与变量说明 .....               | 1-14 |
| § 7 表达式 .....                     | 1-18 |
| § 8 标准函数 .....                    | 1-23 |
| § 9 RT-11 PASCAL 语言的扩充 .....      | 1-25 |
| 习 题 .....                         | 1-28 |
| 第二章 语句 .....                      | 2-1  |
| § 1 输入、输出初步 .....                 | 2-1  |
| § 2 赋值语句 .....                    | 2-10 |
| § 3 复合语句 .....                    | 2-10 |
| § 4 控制语句 .....                    | 2-11 |
| § 5 RT-11 PASCAL 关于语句的扩充与限制 ..... | 2-28 |
| 习 题 .....                         | 2-29 |

|                           |      |
|---------------------------|------|
| 第三章 简单数据类型 .....          | 3-1  |
| § 1 类型概念 .....            | 3-1  |
| § 2 类型定义 .....            | 3-1  |
| § 3 简单类型 .....            | 3-2  |
| § 4 类型相容性与赋值相容性 .....     | 3-8  |
| 第四章 构造数据类型 .....          | 4-1  |
| § 1 集合类型—构造类型 1 .....     | 4-2  |
| § 2 数组类型—构造类型 2 .....     | 4-8  |
| § 3 记录类型—构造类型 3 .....     | 4-24 |
| § 4 文件类型—构造类型 4 .....     | 4-39 |
| 习 题 .....                 | 4-59 |
| 第五章 动态数据结构 .....          | 5-1  |
| § 1 静态变量与动态变量 .....       | 5-1  |
| § 2 指示器类型与动态变量 .....      | 5-2  |
| § 3 勾链表与树 .....           | 5-6  |
| § 4 RT-11 PASCAL的限制 ..... | 5-18 |
| 习 题 .....                 | 5-19 |
| 第六章 过程与函数 .....           | 6-1  |
| § 1 调用过程和定义过程 .....       | 6-1  |
| § 2 改变动作—值参数 .....        | 6-3  |
| § 3 取得结果—变量参数 .....       | 6-5  |
| § 4 计算一个值—函数 .....        | 6-8  |
| § 5 标识符的作用域 .....         | 6-10 |
| § 6 过程与函数的进一步运用 .....     | 6-14 |

|        |                           |       |
|--------|---------------------------|-------|
| § 7    | RT-11 PASCAL 的扩充与限制 ..... | 6-25  |
|        | 习 题 .....                 | 6-26  |
| 第七章    | 程序设计实例 .....              | 7-1   |
| § 1    | 最优路线 .....                | 7-1   |
| § 2    | 语言结构与编译程序 .....           | 7-11  |
| 第八章    | RT-11 PASCAL 系统概况 .....   | 8-1   |
| § 1    | 分布文件 .....                | 8-1   |
| § 2    | 系统工作流程 .....              | 8-2   |
| § 3    | PCL 程序功能 .....            | 8-3   |
| § 4    | 系统先进功能 .....              | 8-3   |
| § 5    | 其它有效的实用程序 .....           | 8-6   |
| 附录 I   | 语法图 .....                 | I-1   |
| 附录 II  | PASCAL 字汇表 .....          | II-1  |
| 附录 III | 参考资料 .....                | III-1 |

## 第一章 Pascal 语言概述

### § 1 程序举例

统计存放在文件 input 中字符的个数, 以及其中空格、逗号和句号个数的 Pascal 程序如下:

```
Program Countchars (input, output);  
Const blank=' '  
      comma=',';  
      period='.';  
var charcount, blankcount,  
     commacount, periodcount: integer;  
     character: char;  
begin  
    charcount:=0; blankcount:=0;  
    commacount:=0; periodcount:=0  
while not eof do  
begin  
    read (character);  
    charcount:=charcount+1;  
    if character=blank  
    then blankcount:=blankcount+1  
    else if character=comma  
    then commacount:=commacount+1
```

```

else if character=period
then periodcount:=period count+1
  end: { while }
writeln(charcount, 'chardctor');
writeln(blankcount, 'blanks');
writeln(commacount, 'commas');
writeln(periodcount, 'periods')
end { countchars }.

```

## § 2 程序结构、语言要素和特点

### 2.1 程序结构

在 Pascal 语言中，程序是由“程序首部”、分号“;”、“分程序”和句号“.”组成。这可用图来描述。这样的图称为语法图，图中矩形框围起来的内容在别处另行定义，而端圆框围起来的是必

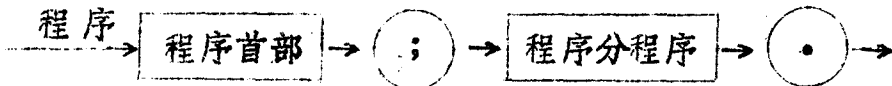


图 1.1

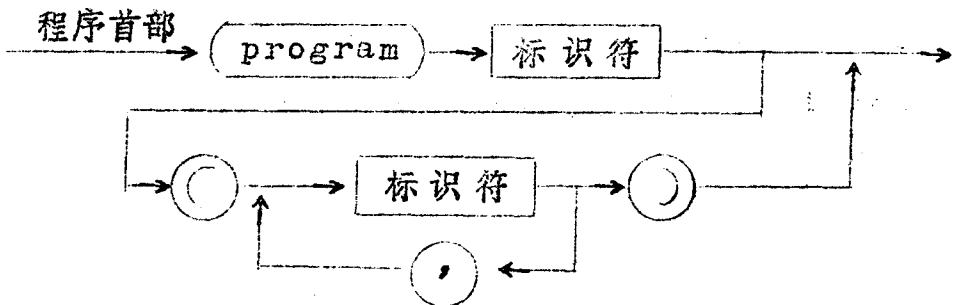


图 1.2

须出现的实际字符，它们是不必进一步定义的对象。下面分别给出程序首部和分程序的语法：

其中紧接在 program 后的标识符是程序的名字。

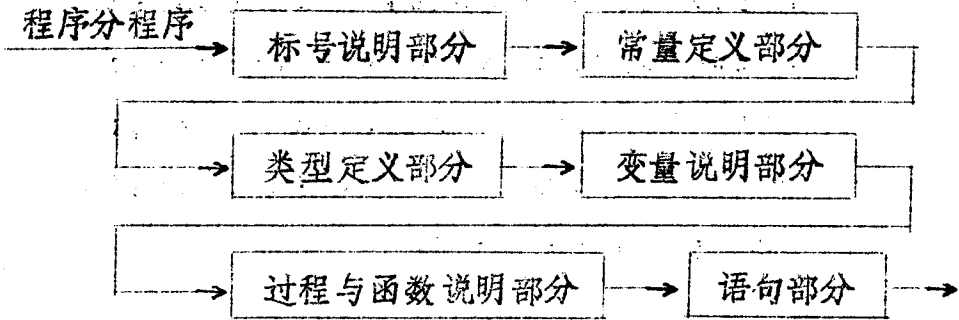


图 1.3

这里，语句部份是一个复合语句。复合语句的语法是

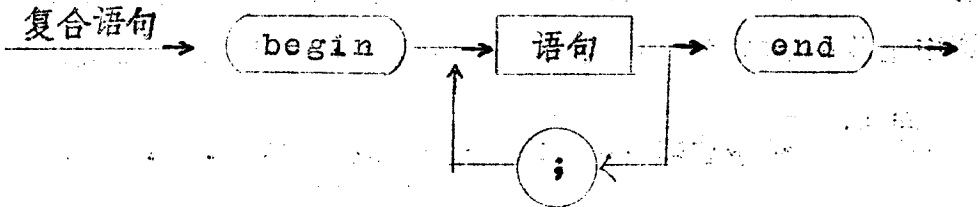
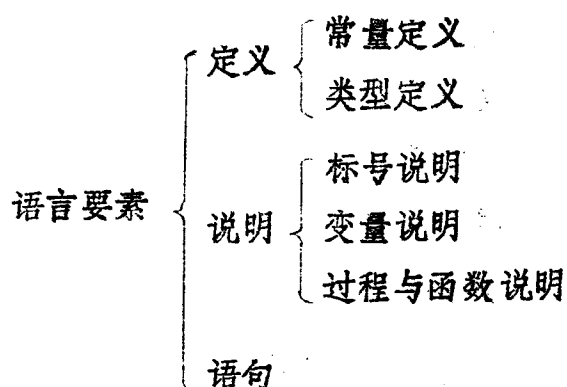


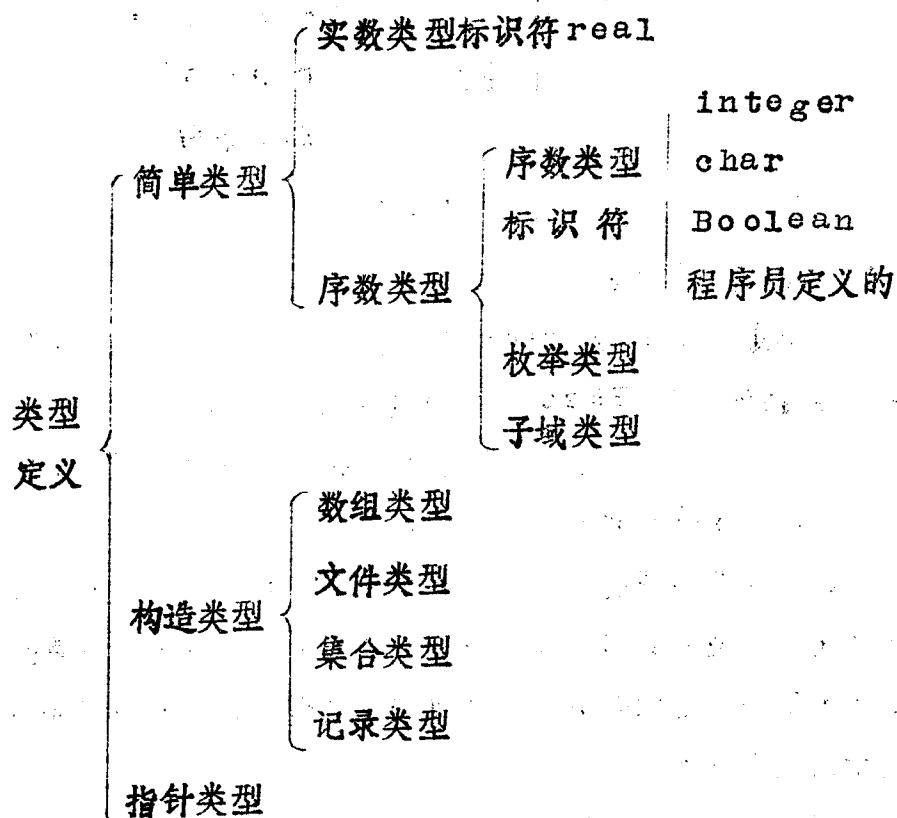
图 1.4

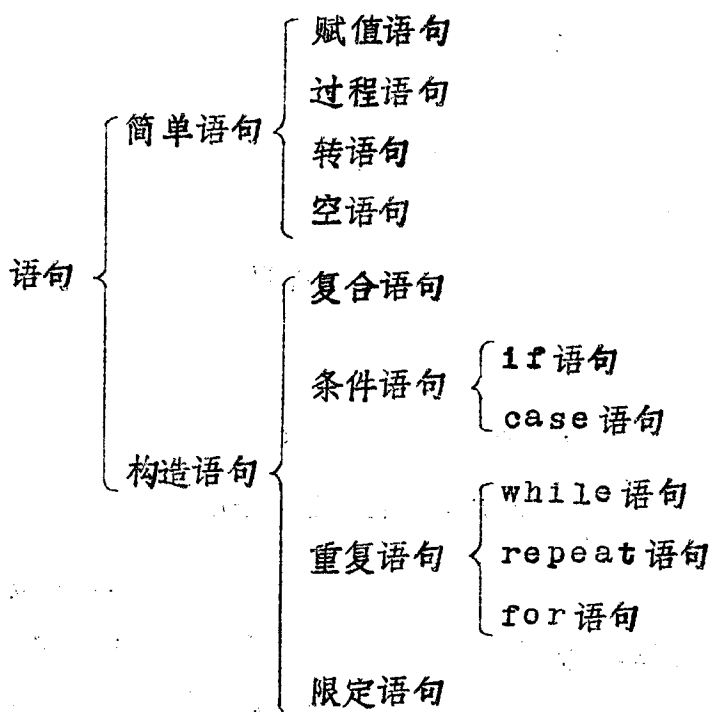
## 2.2 语言要素

Pascal 语言的要素是定义、说明和语句：



对于 Pascal 语言，特别要提出的是：





### 2.3 主要特点

对熟悉Algol 或Fortran的读者，先介绍 Pascal 语言的特点是有好处的。Pascal 语言的主要特点如下：

1. 语句部分中出现的变量（各种各样类型的变量），必须先说明，即使是语句标号也要进行说明。

2. 标准类型有整数类型、实数类型、字符类型、布尔类型、正文文件类型。简单类型还有枚举型和子界类型。构造类型有数组、文件、集合和记录类型。从而可构造复杂的数据，如集合的数组、记录的文件等。

3. 数据可以动态地分配，并由指针进行访问，从而具有表处理的能力。

4. 有多种的控制语句。如 case 语句，它接近Algol 的开

关和Fortran的计算转语句。

5. 形参与实参的结合有如Fortran那样，它没有Algol的换名表达式。

6. 数组的界只允许是常量，因此没有Algol中的变界数组。

7. 如Algol那样，过程和函数可以递归地调用。

8. 与Fortran不同，Pascal中的关键字是保留字，因此不能用作标识符。

9. 与ALGOL相同，语句之间的分隔符为“；”。

10. 与ALGOL不同，复合语句中不可带有任何说明和定义。

### § 3 特殊符号

任何一种程序设计语言都有它自己的一套符号。一般地，符号是由一个或几个字符组成。在Pascal语言中，要用到下列字符：

1. 字母（共26个）

a、b、c、d、e、f、g、h、i、j、k、l、m、n  
o、p、q、r、s、t、u、v、w、x、y、z。

2. 数字（共10个）

0、1、2、3、4、5、6、7、8、9。

3. 其它

+ - \* / = < > . , : ;

↑ ( ) [ ] { } 空格

在Pascal程序中，单个字符并不一定具有确定的含义，但是，它们可组成具有特定意义的符号。如组成标识符、数、字符串和标号等。这些都是Pascal语言中的符号。此外，在Pascal语言

中，还有下列50个特殊符号。为了便于查阅，我们列出这些特殊符号以及它们在Pascal语言中的用法。

#### 1. 非字特殊符号

|     |                  |
|-----|------------------|
| +   | 加                |
| -   | 减                |
| *   | 乘                |
| /   | 除                |
| <   | 小于               |
| <=  | 小于等于             |
| =   | 等于               |
| <>  | 不等               |
| >   | 大于               |
| >=  | 大于等于             |
| .   | 小数点，记录成分选择，程序结束符 |
| ,   | 隔开一个表中的项         |
| :   | 隔开变量名字和类型        |
| ;   | 隔开语句             |
| :=  | 赋值运算符            |
| ..  | 隔开子界类型说明中的上、下界   |
| (和) | 括参数表，括表达式        |
| [和] | 括下标表，集合表达式       |

#### 2. 字符符号(保留字)

|       |         |
|-------|---------|
| and   | 逻辑交     |
| array | 数组类型开始符 |

|                        |                                                                  |
|------------------------|------------------------------------------------------------------|
| <code>begin</code>     | 复合语句开始符                                                          |
| <code>case</code>      | <code>case</code> 语句开始符                                          |
| <code>const</code>     | 常量定义部分开始符                                                        |
| <code>div</code>       | 整除运算符                                                            |
| <code>do</code>        | <code>for</code> 语句, <code>while</code> 语句, <code>with</code> 语句 |
| <code>downto</code>    | <code>for</code> 语句                                              |
| <code>else</code>      | <code>if</code> 语句                                               |
| <code>end</code>       | 复合语句结束符, 记录类型结束符,<br><code>case</code> 语句结束符                     |
| <code>file</code>      | 文件类型开始符                                                          |
| <code>for</code>       | <code>for</code> 语句开始符                                           |
| <code>function</code>  | 函数过程说明开始符                                                        |
| <code>goto</code>      | <code>goto</code> 语句开始符                                          |
| <code>if</code>        | <code>if</code> 语句开始符                                            |
| <code>in</code>        | 集合成员关系运算符                                                        |
| <code>label</code>     | 标号说明部分开始符                                                        |
| <code>mod</code>       | 取余运算符                                                            |
| <code>nil</code>       | 指针变量的值                                                           |
| <code>not</code>       | 逻辑非                                                              |
| <code>of</code>        | 数组、文件和集合类型中                                                      |
| <code>or</code>        | 逻辑并                                                              |
| <code>packed</code>    | 紧缩构造类型开始符                                                        |
| <code>procedure</code> | 过程说明开始符                                                          |
| <code>program</code>   | 程序开始符                                                            |

|        |              |
|--------|--------------|
| record | 记录类型开始符      |
| repeat | repeat 语句开始符 |
| set    | 集合类型开始符      |
| then   | if 语句中       |
| to     | for 语句中      |
| type   | 类型定义部分开始符    |
| until  | repeat 语句中   |
| var    | 变量说明部分开始符    |
| while  | while 语句开始符  |
| with   | with 语句开始符   |

字符在 Pascal 中是保留字，它不能用作标识符。为醒目起见，今后在字符的下面都加以横线（除非用大、小写区别二者）。

#### § 4 标识符

在 §1 中，我们以 blank、comma 和 period 分别表示空格、逗号和句号，以 charcount、...、periodcount 和 character 分别表示字符计数器、...、句号计数器和每次读到的字符。这些名字称为标识符。

标识符是字母开头的字母数字序列。它的语法为：

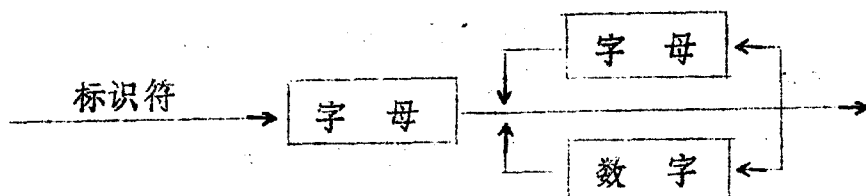


图 1.5