

APPLE III

電腦語言

許 光 智
陳 明 哲

編譯 **PASCAL**



協群科技出版社

APPLE III

電腦語言

PASCAL

許 光 智
陳 明 哲 編譯

協 群 科 技 出 版 社

APPLE III 電腦語言 PASCAL

編譯者：許光智 陳明哲

出版：協群科技出版社

發行：協群科技出版社

香港中環卑利街684號二樓

印刷者：廣源印務局

青山道875號工廠大廈

定價：H.K.\$ 38.00

目 錄

1. 何謂 Apple II Pascal	1
2. 概論	7
3. 單一資料型式	31
4. 式子與指定	47
5. 控制流程	61
6. 程序和函數	77
7. 數列、字組和字串	93
8. 錄目	115
9. 指標和動態變數	127
10. 檔和輸入／輸出	139
11. 一般輸入／輸出	157
12. 其他的輸入／輸出	175
13. 特殊的內程序	185
14. 資料館單元體	197
15. 程序分段	215
附錄 A	
TRANSCEND 和 REALMODES 單元體	225

附錄B

PGRAF 單元體	233
-----------------	-----

附錄C

CHAINSTUFF 單元體	255
----------------------	-----

附錄D

APPCESTUFF 單元體	261
----------------------	-----

附錄E

浮點算術	271
------------	-----

附錄F

APPl e II Pascal 編輯器	293
----------------------------	-----

附錄G

特殊用法	309
------------	-----

附錄H

和 Apple II Pascal 的比較	323
-----------------------------	-----

附錄I

語 法	329
-----------	-----

附錄J

附 表	347
-----------	-----

附錄K

TURTLEGRAPHICS 單元體	361
--------------------------	-----

1**何謂Apple III Pascal**

簡介

Apple III Pascal 是 Apple III 電腦執行 Pascal 語言的工具，它是以 Kathleen Jensen 和 Niklaus Wirth 在「Pascal User Manual and Report」中的原始定義所發展的 UCSD Pascal 為基礎的，本章將就 Pascal 的特性做一詳明的描敘並就其與 Basic 和 Fortran 做一比較。

Pascal 是類似 Algol 的高階語言；也就是說它是由 Algol 語言加入許多高階語言的基本概念發展而成。它具有像 Algol 的自由格式和方格狀結構；同時它更具有多種不同的資料型式和讓設計者去定義新資料型式的能力。

Apple III Pascal 的程式先書寫成一個文字檔，經過系統中的編輯器編譯後，我們稱之為該程式的來源。次由 Pascal 翻譯器加以翻譯後即生成一個類似機器語言的 P-碼檔。

P-碼能經由編譯器的程式而被迅速的執行。這個編譯器程式的名稱是 P-Machine，為軟體系統中的積和部份。使用 P-碼和編譯器程式有一個很重要的優點：較不受限制，Apple III Pascal 程式同樣地可經由不同電腦的 P-machine 來加以處理。

高階語言的設計哲學在於不必考慮電腦本身硬體部份的細節，例如：Pascal 就不必去設計儲存器中的定址工作。整套 Pascal 系統中已經包含了記憶管理的工作，無須由 Pascal 程式中自行設計，同樣地整個系統已經對特殊的 Pascal 所需之機器功能都予以先行規劃設計，省却了 Pascal 程式書寫時的繁複

工作。

Pascal和BASIC的比較

Basic 和 Pascal 和程式設計上有以下數點基本上的不同

。

- 行數碼：Pascal 沒有行數碼，事實上行列的中斷對Pascal 語言無任何意義在 Pascal 中，使用者可將數個指述寫成一行，或將一行中的指述寫成數行，每個指述間的分斷均由分號（；）來隔開。Pascal 的程式在書寫，編輯，修正時均比 Basic 的程式來得簡便，因為它不需要像 Basic 般的去維護行數碼。
- 變數：所有 Pascal 程式中的變數必須先經由宣告它的識別字和資料型式後方能使用。而 Basic 除了數列以外的變數，並不需要宣告。
- 流程的控制：Pascal 對於指敘的執行順序有數種控制的方法。這些方法和 Basic 中的 IF, FOR, GOTO, GOSUB/RETURN 不盡相同。同樣的題目中，Pascal 的程式比 Basic 的程式更易於閱讀和瞭解。
- 程序：Pascal 中的程序（Procedure）只是一個副程式，可經由主程式中的需要而執行作業，它比 Basic 中的 GOSUB/RETURN 更具威力，主程式可供給參數值給程序去執行。
- 函數：和 Basic 所設計的函數是一樣的，然而 Pascal 的函

數定義中的指述在數目上不必像Basic中受到限制。

- 方格結構：方格結構中，函數和程序可視為獨立的個體，它們可具有和主程式相同名稱的變數，如此使得整個程式更簡潔，易於了解Basic在此方面則無法做到。
- 實體位址：在Pascal中沒有POKE，PEEK，CALL等的指令Pascal並不需要實體位址，Pascal的數種機構使之成為不要。

Pascal和FORTRAN的比較

Pascal和FORTRAN有以下值得注意的差別。

- 沒有行數碼。
- 識別字（名稱）受限制較少。
- 程式中的格式較不受限制。
- 程式中的控制結構較受限制。
- 程序和函數可重複，也就是它們可以直接或間接地呼叫自己。
- 方格結構（如上所述）。
- 變數中無對等性。
- 變數在程式執行中可經由指示器來做動態的創造。
- 所有變數均需經過宣告。
- 沒有OWN（SAVE）變數。
- I/O的格式較簡單。

Pascal程式的結構

每一個Pascal 程式均有下列的架構

標頭(Program heading)：包括program 這個字及程式名稱；

宣告：使用者所設計的資料型式，變數等。

定義：程序和函數的設定

BEGIN

由(;) 分隔的指述

END 及一個句號(.)。

例題

以下這個例題，讓你先瞭解 Pascal 程式的結構，此例題將於第二章中為你解釋它的內容。

```

PROGRAM FIRSTEXAMPLE;                { program heading }

VAR I:INTEGER;                        { declaration of
                                     a variable }

PROCEDURE DISPLAY (J:INTEGER);        { procedure definition }
BEGIN
  WRITELN;
  WRITELN('The number is ', J)
END;                                   { end procedure
                                     definition }

BEGIN                                 { begin program body }
  FOR I:=0 TO 10 DO DISPLAY(I)        { statement }
END.                                  { end of program }
```

The output from this program appears on the screen as follows:

```
The number is 0
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9
The number is 10
```

2**概 論**

本章對Apple III Pascal 的主要特色做一介紹，並引導你如何在此書中尋找進一步的資料。

Pascal的文字源

如同英文句子由文字、空白、句點所組成的，Pascal 有一套專屬的符號來組合成它的程式主文，這些符號是：

- 保留字 (Pascal 的固定字彙)
- 識別字 (部份由使用者設計)
- 數字常數 (程式中之數字做為資料使用)
- 字元及字串常數 (程式中之文字做為資料使用)
- 定界值 (特殊文字及標點符號)

Pascal 程式中亦可加入註解，這些註解將不經由編譯器處理，如此可使閱讀者易於了解。凡在特殊符號{ }及(* *)中者均屬於註解，例如：

```
{This is a comment}
(*This is another comment*)
```

Pascal 程式是自由格式型態，設計者可隨意的把空白加入程式，或者將程式寫成鋸齒狀，例如下列這個Pascal 的指述

```
FOR I:= 0 TO N DO BEGIN A:=I; B:=2*I+1 END
```

可改寫為

```
FOR I := 0 TO N DO BEGIN
    A := I;
    B := 2*I + 1
END
```

```
For i := 0 to n do Begin a:=i; b:=2*i + 1 End
```

或者其它的型態都可以，祇要把符號的順序排列好即可。在上面的例子中，英文字均以大寫表示，實際上並無此限制，用小寫字母書寫亦無妨，（如在文字常數時則是例外）。

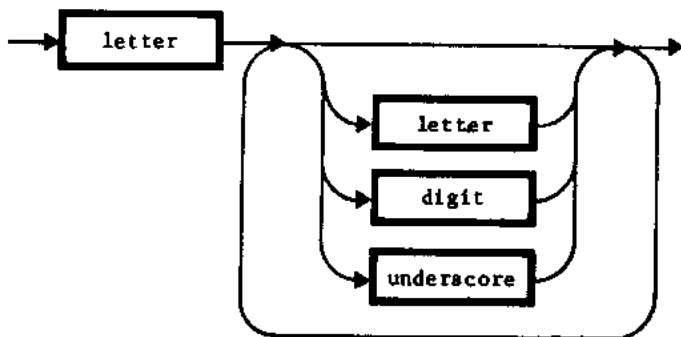
符號

保留字

保留字係 Pascal 中具有固定意義的基本字彙如 FOR , WHILE, AND, DO, BEGIN 等。保留字請參閱附錄 J。

識別字

識別字可經過使用者之設計而建立變數，資料型態，程序等的名稱，在經宣告後而有其意義。部份的識別字已經建立於 Pascal 語言當中，不必再經宣告即可使用，識別字之語法如下：



每個識別字的第一個字母必須為英文字母，以後則可接字母，數字或底線符號（`_`）。英文字母為大寫或小寫對編譯器而言均是相同的。（`_`）在識別字中毫無意義，其目的僅在使識別字易於辨讀。注意，僅有前 8 個字元有用（不包括底線符號（`_`）），下列例子均是同樣的識別字其間並可互換。

<code>MYNUMBER</code>	<code>mynumber</code>
<code>MY_NUMBER</code>	<code>MY_Number</code>
<code>MY_NUMBER_VALUE</code>	<code>MY_NUMBER_SYMBOL</code>

另外有一重要的限制：識別字不可寫成和保留字相同。

數字常數

數字常數在 Pascal 程式中均被視為資料來處理，它可分為整數和實數兩種，整數常數的範圍為 -32768 到 32767。下列數例均為整數：

`-1 2 865 16383 -2000 0 1949`

實數常數則可為帶小數點或以 E 代表 10 次方的指數乘幕。下列數例均代表等值的數字：

`3.14159 0.314159E1 3141.59E-3 31415.9E-4`

字元常數

在 Pascal 中表示資料型式為字元，而且是單獨的一個字元。用引號括起來表示即可，例如：

'A' 'a' 'Ø' '+' ''''

'A' 和 'a' 分別表示字元 A 及 a，二者並不相同。在字元中大寫和小寫字母並不相同，而 '''' 則表示 ' 這個字元，切記勿使用雙引號"，此符號在 Pascal 無任何意義。

字串常數

如同字元常數，字串常數係由好幾個字元連串而成，亦是當為資料來處理。例如：

'Smith' '\$408.23' 'Type your name: ' 'DON'T WORRY' ''

同樣地，大寫及小寫字母在此亦是不同的字元，最後一例即表示該字串中無任何字元。

定界值

它做為各符號間的分隔訊號，否則各指述間必須有空白或跳行以表示分隔。

這些特殊的字元（亦有二字元組合者）有數種不同的意義，如運算子數列註標，註解標示等，單字元的定界值有：

, . ; : ' () [] + - / * = < > ^ { }

二字元的定界值有：

:= .. (* *) <= >= <>

宣告

一個 Pascal 程式可視為二個不同的段落，指述和宣告，指述用於程式中的執行，而宣告則在於表明識別字的性質。

變數宣告

所有變數均須事先宣告（除了動態變數以外），變數宣告提供了該識別字的名稱，資料型態，及儲存的位址，變數的宣告由保留字 VAR 來執行，例如：

```
VAR NEWVALUE: INTEGER;  
    RESULT: REAL;
```

上例中宣告了二個識別字 NEWVALUE 和 RESULT，前者是整數變數，後者為實數變數。

常數宣告

識別字亦可宣告為數字或字元常數，常數識別字可在程式中表示其為該常數，常數的宣告由保留字 CONST 來執行，例如：

```
CONST PI=3.14159;  
      ROWSIZE=64;  
      COLUMNSIZE=2048;
```

上例中宣告了三個常數識別字 PI, ROWSIZE, and