



## PASCAL への勧誘\*

和田 弘\*\*

### まえがき

プログラミング言語として PASCAL を、研究者はもちろん学生の教育に用いることを提唱したい。

わが情報処理学会は IFIP の設立に呼応して創立されたものであり、1960 年のことである。当時のプログラム言語はアセンブラの時代であり、コンパイラ言語として話題に上っていたのは IBM の FORTRAN と IFIP の ALGOL 60 だけであった。筆者が Fortran についての J. W. Backus の論文を読んで、数式がそのまま代入文になることを知った時は感心した。これができるのであれば、DO 文、IF 文は当然のことと受取った。当時はまだ IBM の計算機も多くなかったから、Fortran もそれほど普及してはいなかった。

これに対して、Algol は動的な activation の要領が判りにくかったり、慣れを必要とするものではあったが、自由に討議できる算法定語であるから、学問の場ではこれを使うのが適当であろうと考えられた。そこで、学会では Algol を勧めるべきだとして、学会誌に“Algol のページ”を設けて、その普及を図ることになったのである。

### 現 況

その後計算機の普及とともに Fortran が次第に用いられるようになったが、その一因としては東京大学に 1965 年に設けられた共用の計算センターが、ハードウェアとしては国産機を採用したのに、ソフトウェアとしてはもっぱら Fortran によって教育と研究をしたこともあげられよう。事実、入出力操作は人手によるし、主な利用対象は数値計算であるという時期に生まれた Fortran はその目的に十分叶うものをもっていた。その後言語についての研究が進み、それらの

成果を取り入れた言語はいくつか登場したが、IBM と Fortran との力は強く、現在に至っている。

一方、実業界に計算機が使われるようになって、データの記述法にも不満が出て、CODASYL の手で Cobol が作られて以来、わが国でも現在最も広く使われる言語となっている。

現にこれらの 3 言語は ISO でも標準として採用しており、国内でも JIS になっている。しかし、われわれはこれらの言語をどうすることもできない。改訂などの実質的決定権者は IBM であり、CODASYL であるといっても過言ではない。

### 当面の課題

計算機のハードウェアは物理学の進歩を利用して LSI まで進み、その価格は低下するばかりであるが、反面ソフトウェアはますます大きくなるばかりで、システムでその占める価格は巨大なものになって来た。これを改善するためいろいろな資料なり対策が提案されて来ているが、その中で比較的広く同意が得られて実施策が講ぜられているのは、広義に structured programming といわれるものであろう。プログラム言語とプログラム作成法とは密接な関係があると考えられるが、この提唱者 E. W. Dijkstra は Algol 系の考えに立つ人で、本人もあるいは Algol を広く使っているヨーロッパの人々にも、プログラムの作成法として特に新しいことが述べられたとは思わないとも考えられよう。

実用面では Fortran, Cobol に押されて欧州以外では影が薄くなったかに見える Algol 60 は、以降地味な研鑽によっていくつかの変遷があった。

まず Algol 68<sup>1)</sup>は Algol 60 を一般化したものであるが、その程度が過ぎるようであり、柔軟すぎて難解であり、コンパイラも大変であろう。

Algol 60 自身は最近 IFIP によって Modified version<sup>2)</sup>が出された。解釈によってコンパイラ間に差異がある現状を防止することが狙いらしい。ISO は

\* An Invitation to the PASCAL by Hiroshi WADA (Faculty of Engineering, Seikei University)

\*\* 成蹊大学工学部

IFIP から現在の Algol の標準を何とかするように求められている。

Algol 系のシミュレーション言語としての Simula 67<sup>9)</sup>は、Algol でのブロックはその操作が終了とデータが抹消されるのを防いだものとして特色がある。

## PASCAL

ここで主題にする PASCAL<sup>4)</sup>は N. Wirth の提案である。彼は C. A. R. Hoare とともに IFIP-WG 2/1 の幹事として Algol 60 の改良案<sup>5)</sup>を示したり、IBM/360 について PL 360<sup>6)</sup>を発表して、S/360 の命令に批判を加えたこともある。PASCAL は Algol 60 を基に、Algol 68 と反対の方向、即ち簡素なものへと改定したのである。

以下 PASCAL がよいと考えられる理由を述べる。

- 1) 文法が極めて簡素であるから、全体を把握するのが易しい。
- 2) C. A. R. Hoare とともに公理系による厳密な言語の定義<sup>7)</sup>が与えられている。
- 3) データ構造が豊富で、かつプログラマは data type を設けることができる。従って科学技術用にもビジネス用にも使える。
- 4) コンパイルの際に data type を check できる。
- 5) ブロック構造であるから、データは必要に時だけメモリにあり、メモリの節約ができる。また抽象化の考えに従ってプログラムを段階的に記述して行くに当って recursive を含められる。
- 6) プログラムを作成して行く過程で、data type が statement の種類と同じ思想で展開できる。
- 7) コンパイラは大きくなく、プログラム・エラーは全部一度に報告される<sup>8)</sup>。
- 8) 入出力文も整っている<sup>9)</sup>。

従って、プログラマは自分の意図するところを structured programming 的に判りやすく作って行けるし、出来たものは読みやすく、誤りの少ないものになっているような素質を備えている。PASCAL を換言すれば、著書 Structured Programming<sup>9)</sup>の中で、C. A. R. Hoare が "Notes on Data Structuring" で算法言語として提案したものをプログラム言語として実現したものと解せよう。

さらに言えば、

- 9) 広い意味で extensible である。例えば P. B. Hansen は Simula の class を1つの data type として Pascal に加えて Concurrent Pascal<sup>10)</sup>を

提案し、それでいくつかの OS を作った実績<sup>11)</sup>を報告している。Wirth 自身も同じ趣旨で言語 Modula<sup>12)</sup>を試作している。

現在メーカーの苦心している OS は多彩な入出力装置の出現に依ると推察するが、これらを用いれば記述が大変楽になり、しかもエラーをコンパイラに指摘してもらえる。

マイクロコンピュータは計算機の従来の利用範囲とかなり違う分野にも出ようから、それらの用途に適した OS は、従来のパターンから外れたものも必要になる。今後提案される OS 用の記述言語は、PASCAL が基礎になると察せられる。

## むすび

大きなライブラリをもつユーザとメーカーは使用言語を急に変更することはできないとしても、わが国の課題であるプログラミングについての自主技術を確立するには、大学が自らの研究にも、学生に入門を教えるにも、よい言語を採用することが長期計画の第一歩であろう。幸い、東大<sup>13)</sup>を始めいくつかの大学で Pascal のコンパイラが動いているようだ。この方向の活動を強化するには協力が必要である。学会(ことに理事会)は企業法人のようなことを考えるのではなく、こういうことを検討する場になって貰いたい。

## 参考文献

- 1) V. Wijngaarden: Report on ALGOL 68, Math. Center, Amsterdam. (1968)
- 2) R. M. de Moran et al.: Modified Report on ALGOL 60, Computer Journal. Vol. 19, No. 4, pp. 364~379 (1976)
- 3) O. J. Dahl & C. A. R. Hoare: Hierarchical Program Structures, Structured Programming, Academic Press, London (1972)
- 4) N. Wirth: Systematic Programming, pp. 155-162, Prentice-Hall, N. J. (1973)  
和田英一: PASCAL の文法, bit Vol. 8, No. 4, pp. 341-366, (1976)
- 5) N. Wirth & C. A. R. Hoare: A contribution to the development of ALGOL, Comm, ACM. Vol. 9, No. 6, pp. 413-431 (1966)
- 6) N. Wirth: PL 360 Jour. ACM Vol. 15, No. 1, pp. 37-74, (1968)
- 7) C. A. R. Hoare & N. Wirth: Axiomatic definition of PASCAL, Acta Inform. Vol. 1, 2, (1973)
- 8) N. Wirth: Design of a PASCAL compiler, Software-practice & experience, Vol. 1, pp. 309-333 (1971)

- 9) K. Jensen & N. Wirth: PASCAL: user manual and report, Springer, Berlin (1974)
  - 10) P.B. Hansen: Concurrent PASCAL, Trans. Software Engg. IEEE, Vol. 1. No. 2, pp. 195-207 (1975)
  - 11) P.B. Hansen: The Solo operating system, Software-practice & experience, Vol. 6, pp. 141-205 (1976)
  - 12) N. Wirth: Modula, Software-practice & experience Vol. 7, pp. 3-84 (1977)
  - 13) 和田, 武市: PASCAL を使用してみた, 学会16回大会 p. 61 (1975)
  - 14) 武市, 疋田, 安村: 東大の PASCAL コンパイラ, 学会16回大会 62 (1975)
  - 15) 疋田輝雄: コンパイラのキットを用いた PASCAL の移植, 日経エレクトロニクス, No. 12, 13, pp. 100-131 (1976)  
(昭和52年6月13日受付)
-