

# BASICからFÖRTRANへの 変換支援プログラム

金原慎一\*

パーソナルコンピューター(パソコン)を用いて技術計算用プログラムを作成する場合、選択の対象となる言語は、現時点ではBASICとFÖRTRANであろう。

BASICでは、ステートメントの編集を文番号を用いて行うのでプログラミングが比較的容易である。また、コンパイルを必要としないので直ちにテストランを行うことができ、デバッグも容易であるが、反面、ユーザーエリアが小さく、一般に演算速度も遅い。

一方、FÖRTRANはユーザーエリア、演算速度の点ではBASICを遙かに上回っており、構造解析等の技術計算には大変有利である。しかし、実行前にコンパイルを行う必要があり、かつパソコンのコンパイラーの速度が非常に遅いので、プログラムのデバッグに相当な時間を必要とする。またFÖRTRANソースプログラムは通常、OSコマンドであるエディターによって作成するが、このコマンドによるプログラムの修正、編集はBASICのそれと比べて大変複雑である。

従って、FÖRTRANのプログラムを作成する場合、あらかじめBASICプログラムを作成し、これによってロジックのチェックを行った後にこれをFÖRTRANに変換することが出来れば大変便利である。本文で紹介するのは、上記の手順のうち、BASICソースファイルをFÖRTRANソースファイルに変換する部分に用いるプログラムである。

表-1はBASICとFÖRTRANのコマンド及び組み込み関数の対比を示したものである。

本プログラムは、BASICソースファイルをFÖRTRANソースファイルに変換する際に、修正漏れを無くすること、及び修正作業にあたって面倒なエディターを使用しないことに主眼を置いて作成したものである。このプログラムを実行すると、まず表中の※印を付したコマンドを自動変換した新規のファイルを作成する(中

間ファイルと呼ぶ)。次に、この中間ファイルの内容をプリンターに出力するとともに、自動変換が不可能なコマンドがある場合はそれも含めて出力する(図-1参照)。

表-1. 主なコマンド及び組み込み関数の対比

項目	BASIC	FÖRTRAN	
宣言文	DIM	DIMENSION	※
	DEFINT, DEFSNG, DEFDBL, DEFSTR	IMPLICIT	
	% .! .# . \$		
四則演算	+, -, *, /	+, -, *, /	
べき乗	^	**	※
入力	INPUT, LINE INPUT	READ(*, nnnn)	
出力	PRINT, PRINT USING	WRITE(*, nnnn)	
	LPRINT, LPRINT USING	WRITE(6, nnnn)	
Sub. Pro.	nnnn	SUBROUTINE XXXXXX	
	GOSUB nnnnn	CALL XXXXXX	
	ON i GOSUB j, k, l, ..., m	-----	
Jump	GOTO nnnnn	GO TO mmm	※
	ON i GOTO j, k, l, ..., m	GO TO (j, k, l, ..., m), i	
IF	A=B, A>B, A<B	A.EQ.B, A.GT.B, A.LE.B	
	A<B, A<=B	A.LT.B, A.LE.B	
	THEN, ELSE, ELSE IF	THEN, ELSE, ELSEIF(e)THEN	
関数定義	DEF FNxxx(y, z)=	xxx(y, z)=	※
Root	SQR(X), SQR(X#)	SQRT(X), DSQRT(X)	※
余弦	COS(X), COS(X#)	COS(X), DCOS(X)	
正弦	SIN(X), SIN(X#)	SIN(X), DSIN(X)	
正接	TAN(X), TAN(X#)	TAN(X), DTAN(X)	
逆正接	ATN(X), ATN(X#)	ATAN(X), DATAN(X)	※
絶対値	ABS(X), ABS(X#)	ABS(X), DABS(X)	
指数関数	EXP(X), EXP(X#)	EXP(X), DEXP(X)	
自然対数	LOG(X), LOG(X#)	LOG(X), DLOG(X)	
Loop	FOR i=n TO m[ Step j]	DO kkkk i=n, m[ STEP j]	
コメント	REM,	C	※
文の区切	:	-----	※

\* 技術本部設計部設計第二課

この中間ファイルはBASIC上にロードできるので、先に出したりリストを見ながら容易に修正(変換)することができる。こうして変換作業の終わったソースプログラムをASCII SAVEし、このソースファイルにおける各ステートメントの先頭8カラムを取り去ればFÖRTRANソースファイルとして取り扱うことができることになる。

これとは反対に、FÖRTRANソースファイルの先頭に文番号を付け加えることにより、中間ファイルとしてBASIC上にロードすることができる。この、中間ファイル⇄FÖRTRANソースファイルの変換プログラムをも用意してあることは言うまでもない。

```

1000 ◆◆◆ TEST PROGRAM ◆◆◆
1010 DIM Z(3)
1020 INPUT "A=",A
1030 INPUT "B=",B
1040 INPUT "C=",C
1050 INPUT "D=",D
1060 Z(1)=A^2*EXP(B):Z(2)=D*SQR(C):Z(3)=ATN(D)*C^B
1070 FOR I=1 TO 3
1080 ON I GOTO 1090,1110,1130
1090 LPRINT USING "Z(#)=####.#####";I,Z(1)
1100 GOTO 1140
1110 LPRINT USING "Z(#)=#.#####^";I,Z(1)
1120 GOTO 1140
1130 LPRINT USING "Z(#)=####.#####";I,Z(1)
1140 NEXT I
1150 END

```

(a) BASICプログラム



```

◆◆◆ BASIC → FORTRAN TRANSPOSER [ TEST .BAS → TEST .MID ] ◆◆◆
Col. / 1 67 10 20 30 40 50 60 70
+-----+-----+-----+-----+-----+-----+-----+
10000 ' C ◆◆◆ TEST PROGRAM ◆◆◆
10010 ' 1010 DIMENSION Z(3)
10020 ' 1020 INPUT "A=",A
      ' ★★ ★ 《 INPUT 》 がある! ★★ ★
10030 ' 1030 INPUT "B=",B
      ' ★★ ★ 《 INPUT 》 がある! ★★ ★
10040 ' 1040 INPUT "C=",C
      ' ★★ ★ 《 INPUT 》 がある! ★★ ★
10050 ' 1050 INPUT "D=",D
      ' ★★ ★ 《 INPUT 》 がある! ★★ ★
10060 ' 1060 Z(1)=A**2*EXP(B)
10070 ' Z(2)=D*SQR(C)
10080 ' Z(3)=ATAN(D)*C**B
10090 ' 1070 FOR I=1 TO 3
      ' ★★ ★ 《 FOR 》 がある! ★★ ★
10100 ' 1080 GO TO (1090,1110,1130),I
10110 ' 1090 LPRINT USING "Z(#)=####.#####";I,Z(1)
      ' ★★ ★ 《 LPRINT 》 がある! ★★ ★
10120 ' 1100 GO TO 1140
10130 ' 1110 LPRINT USING "Z(#)=#.#####^";I,Z(1)
      ' ★★ ★ 《 LPRINT 》 がある! ★★ ★
10140 ' 1120 GO TO 1140
10150 ' 1130 LPRINT USING "Z(#)=####.#####";I,Z(1)
      ' ★★ ★ 《 LPRINT 》 がある! ★★ ★
10160 ' 1140 NEXT I
      ' ★★ ★ 《 NEXT 》 がある! ★★ ★
10170 ' 1150 END

```

(b) 中間ファイルのプリントアウト

図-1. 変換例