

---

```
DO ;  
DO 指数名 = 初期値 TO 終了値 [BY 増分]  
あるいは  
DO 指数 = 初期値, 終了値 [, 増分 ]
```

---

機能 :

DOはループあるいはグループになったステートメントの始まりを示します。ループあるいはグループになったステートメントはENDDO ;ステートメントで終わらなければいけません。

使用法 :

DOステートメントの最初の形(引数なし)は、IFステートメントの後でTHENあるいはELSE文のステートメントブロックの始まりを示すのに主として用います。

DOステートメントのもう一つの形は、多くのプログラム言語にあるような一般的なループを示します。TSPはDO ... とINDDOステートメントの間のステートメントをDOステートメントに与えられた情報によって示されているだけの回数だけ繰り返して実行します。インデックスあるいはカウンター変数は、一回目は初期値に等しくセットされますが、次からは最終値に達するかそれを越えるまで増分だけ変化します。このテストはループの最後で行われますので、プログラムは常に一回は実行されます。

初期値、最終値それに増分は、整数しか使えなかった初期のTSPとは異なり(正あるいは負の)どんな実数であってもかまいません。増分が負ならもちろんそのインデックスはその絶対値だけ減少しますので、初期値は最終値より大きくなってはいけません。

インデックス変数ループの中で毎回更新されるので、計算の中や添え字として使えます。しかしTSPのDOループはあまり効率のよいプロシジャーループではありませんから、大きなDOループで大量の変数変換や計算をするのは注意した方がよいでしょう。系列の累積和を求めたい場合には、次のようにしてダイナミックなGENRを使えます :

```
ACSUM = X ;  
SMPL 2,N ;  
ACSUM = ACSUM(-1) + X ;
```

あるいは MSD(NOPRINT) X ; (結果は@SUMにあります)

あるいは INPROD X C SUM ;

とすると簡単に得られます。

IFのあるループはGENRの右辺に論理式を置くか、SMPLIFとするのが最も良いでしょう。

例 :

```
DO I = 2 TO 7 BY 1 ;  
SET IM1 = I-1 ;  
SET X(I) = X(IM1) + X(I) ;  
ENDDO ;
```

```
DO I = 2 TO 7 ;  
  SET IM1 = I-1 ;  
  SET X(I) = X(IM1) + X(I) ;  
ENDDO ;
```

この2つの例は、増分の既定値が1ですので同じ結果になります。

```
OLSQ Y C X1 X2 ;  
IF ABS(@DW-2)>.5 ; THEN ; DO ;  
  AR1 Y C X1 X2 ;  
  FORM EQ1 ;  
ENDDO ;  
ELSE ; FORM EQ1 ;
```

この例は、OLSを実行し、ダービン・ワトソン統計値をチェックし、ダービン・ワトソンが2から十分離れていれば同じ式をAR1で実行します。DO ... ENDDOステートメントはダービン・ワトソン検定に失敗したら実行されるステートメントをはさんでいます。