

Fortran 文法のまとめ

プログラムの構造(例)

```
PROGRAM REIDAI
  IMPLICIT NONE
  INTEGER :: I, J, K    !整数型の宣言
  REAL    :: A, B, C    !実数型の宣言
  (実行部分)
END PROGRAM REIDAI
```

入出力(例)

```
WRITE(*,*) I,J,K      !整数の出力
READ(*,*) I           !整数 I の読み込み
WRITE (*,*) 'I=', I  !I=という文字に続けて I の値を出力
```

書式付出力(例, I=13, A=3.14, B=1.234 とする)

```
WRITE(*, '(A,3X,I3,3X,E15.8,3X,E15.8)') 'I=',I,A,B
```

出力結果

```
I=   13   0.31400000E+01   0.12340000E+01
```

これは、次のようにしても同じ結果となる。

```
WRITE(*, '(A2,3X,I3,3X,E15.8,3X,E15.8)') 'I=',I,A,B
```

あるいは

```
WRITE(*, '(1A2,3X,1I3,2(3X,E15.8))') 'I=',I,A,B
```

一般に、A, I, E 等の前に数字を書いた場合、その数字はその欄の個数を意味する。文字の後ろは欄の幅（文字数）であるが、X の後ろには数字を書かない（X の前に数字を書いて X の個数を指定すればすむから。）

一般には、mAn, mIn, mX, mEn.k, m(書式の並び)のように前に個数 m, 後ろに幅 n を指定する。小数部分がある書式では、ピリオドの後ろに小数部分の桁数 k を指定する。

繰り返し(例)

```
DO I=1,100
  A(I) = I+1
END DO
DO I=1,10
  DO J=1,20
    S=S+A(I,J)*B(J)
  END DO
END DO
```

条件判断(例)

```
IF(M == 1 .AND. N /= 0) THEN
  IF(K > 0) THEN
    A = A + 1
  ELSEIF(K < 0) THEN
    A = A - 1
  ELSE
    A = 0
  END IF
END IF
```

サブルーチン, 関数

まとまった処理を別のプログラムとして作成し、必要に応じて呼び出して(CALL分を用いて) 使う.

サブルーチン: 値を変数で渡し, 計算結果は変数で返す.

関数: 値を変数で渡し, 結果は関数名で返す.

(例: 2つの数の和を求めて変数で返すサブルーチン)

```
SUBROUTINE SUMSUB(A,B,C)
  IMPLICIT NONE
  REAL, INTENT(IN) :: A,B           !実数 A,B を入力専用で宣言
  REAL, INTENT(OUT) :: C,D          !実数 C,D を出力専用で宣言
  C=A+B
  D=A-B
```

```
RETURN
END SUBROUTINE SUMSUB
```

呼び出し方：

.....

```
X=2.3
```

```
Y=3.1
```

!X と Y の和と差が, W=4.4, S=-0.8 として戻る.

```
CALL SUMSUB(X,Y,W,S)
```

```
WRITE(*,*) 'X+Y=',Z
```

```
X=1.1
```

```
Y=2.2
```

!今度は X と Y の和と差が, W=3.3, S=-1.1 として戻る.

```
CALL SUMSUB(X,Y,Z)
```

.....

戻ってくる値がただ一つの場合は関数が便利. 関数の定義の仕方は次のようになる.

```
FUNCTION SUM(A,B)
```

```
IMPLICIT NONE
```

```
REAL, INTENT(IN) :: A,B      !実数 A,B を入力専用で宣言
```

```
REAL :: SUM                  !関数値 SUM を実数として宣言
```

```
SUM=A+B
```

```
RETURN
```

```
END FUNCTION SUM
```

関数の場合は, 関数名が出力となり出力専用であることは自明であるから,
REAL, INTENT(OUT) :: SUM とはする必要がないし, するとエラーとなる.

配列の宣言

```
INTEGER :: A(10), B(5,5) !A は 1次元の配列, B は 5×5 の 2次元配列
```

```
REAL :: X(10,10,10), Y(20,10)
```

```
REAL :: Z(0:10,0:10) !番号を 0 から 10 で指定. 11×11 の 2次元配列
```

```

配列のサイズを後で確保するとき（配列を実行時に動的に確保するとき）
REAL, ALLOCATABLE :: A(:, :) !サイズは後で ALLOCATE 文で指定
INTEGER :: SIZE !配列 A のサイズとする.
WRITE(*,*) '配列 A のサイズを入力してください. '
READ(*,*) SIZE
ALLOCATE(A(SIZE, SIZE)) !A を SIZE×SIZE の 2 次元配列として確保.
A=0.0 !A の各要素をすべて 0.0 に設定.
. . . . .
DEALLOCATE(A) !A をメモリーから解放してシステムに返す.

```

以上の事項をすべて盛り込んだプログラム例を以下に示すので、よく研究すること。

問題 3次元空間内の複数の点の座標を読み込み、それぞれの点の原点からの距離とその最大値、最小値、平均、標準偏差を計算するプログラムを作れ。

各点の座標と、距離、には適切な配列を用いる。距離は関数副プログラム、最大値、最小値、平均、標準偏差はサブルーチンで計算することにする。データを a_i , ($i=1, \dots, n$) とすると、平均 m と標準偏差 σ は、次式のようなになる。

$$m = \frac{1}{n} \sum_{i=1}^n a_i, \quad (i=1, \dots, n), \quad \sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (a_i - m)^2}, \quad (i=1, \dots, n)$$

計算の手順は次のようになる。

1. 点の個数の入力
2. 各点の座標の入力
3. 各点の原点からの距離の計算
4. 各点の原点からの距離の最大値、最小値、平均、標準偏差の計算
5. 結果の出力

これを、Fortran のプログラムにすると、以下のようになる。

PROGRAM EXAMPLE

IMPLICIT NONE

INTEGER :: N !点の総数

REAL, ALLOCATABLE :: p(:, :), r(:) !各点の座標と原点からの距離を格納する動的配列

REAL :: rmax, rmin, rmean, rsigma !距離の最大値, 最小値, 平均, 標準偏差

INTEGER :: i, j, istat

INTERFACE

!座標から距離を計算する関数副プログラム

FUNCTION DISTANCE(x, y, z)

IMPLICIT NONE

REAL :: DISTANCE !関数の戻り値

REAL, INTENT(IN) :: x, y, z !x 座標, y 座標, z 座標

END FUNCTION DISTANCE

!一般のデータを格納した配列から, 最大値, 最小値, 平均値, 標準偏差を計算して返すサブルーチン

SUBROUTINE VALUES(a, n, amax, amin, amean, asigma)

IMPLICIT NONE

INTEGER, INTENT(IN) :: n !データの個数の仮引数

REAL, INTENT(IN) :: a(n) !読み込まれたデータ配列の仮引数

REAL, INTENT(OUT) :: amax, amin, amean, asigma !データの最大値, 最小値, 平均, 標準偏差の仮引

数

END SUBROUTINE VALUES

END INTERFACE

WRITE(*, '(A)') '点の総数は?'

READ(*, *) N

```

!配列用のメモリの確保
!p(i,1)は点p(i)のx座標, p(i,2)は点p(i)のy座標, p(i,3)は点p(i)のz座標とする.
ALLOCATE( p(N,3), r(N), STAT=istat ) !配列を動的に確保
IF(istat /= 0) THEN
    WRITE(*,*) '座標と距離を格納する配列のメモリの確保に失敗しました. '
    WRITE(*,'(I3)') '戻り値=', istat
END IF

!座標の読み込み
DO i=1,n
    WRITE(*,'(A,I2,A)') '点', i, ' の x,y,z 座標を入力してください. '
    READ(*,*) (p(i,j), j=1,3)
END DO

!原点からの距離の計算
DO i=1,n
    r(i) = DISTANCE(p(i,1), p(i,2), p(i,3)) !x, y, z 座標を入れて距離を返す関数
END DO

!距離の最大値, 最小値, 平均, 標準偏差の計算 (ここではサブルーチンを用いる)
CALL VALUES(r, n, rmax, rmin, rmean, rsigma)

!結果の出力
!各座標と原点からの距離の出力
WRITE(*,'(7X,A1,15X,A1,15X,A1,11X,A8)') 'X', 'Y', 'Z', 'DISTANCE'
DO i=1,n

```

```

        WRITE(*,'(3(E15.8,1X),E15.8)') (p(i,j),j=1,3), r(i)
    END DO
    WRITE(*,'("原点からの距離の最大値  :",E15.8)') rmax
    WRITE(*,'("原点からの距離の最小値  :",E15.8)') rmin
    WRITE(*,'("原点からの距離の平均値   :",E15.8)') rmean
    WRITE(*,'("原点からの距離の標準偏差:",E15.8)') rsigma

    DEALLOCATE( p, r )    !配列を解放

    STOP
END PROGRAM EXAMPLE

!座標から距離を計算する関数副プログラム
FUNCTION DISTANCE(x,y,z)
    IMPLICIT NONE
    REAL :: DISTANCE          !関数の戻り値
    REAL, INTENT(IN) :: x,y,z  !x座標,y座標,z座標
    DISTANCE = SQRT(x*x+y*y+z*z)
    RETURN
END FUNCTION DISTANCE

!一般のデータを格納した配列から、最大値、最小値、平均値、標準偏差を計算して返すサブルーチン
SUBROUTINE VALUES(a,n,amax,amin,amean,asigma)
    IMPLICIT NONE
    INTEGER, INTENT(IN) :: n    !データの個数の仮引数
    REAL, INTENT(IN):: a(n)     !読み込まれたデータ配列の仮引数

```

```
REAL, INTENT(OUT) :: amax, amin, amean, asigma !データの最大値, 最小値, 平均, 標準偏差の仮引数
real :: s !合計を格納する変数
INTEGER :: i
```

```
IF(n <= 0) THEN
    WRITE(*,*) 'データの個数が 0 以下で計算できません. 終了します. '
    STOP
END IF
```

!データの個数が 1 個の時はとりあえず次のようになる.

```
amax = a(1)
amin = a(1)
amean = a(1)
asigma = 0.0
```

!データの個数が 2 以上の時

```
IF(n > 1) THEN
    !最大値と最小値の計算
    DO i=2,n
        IF(a(i) > amax) THEN
            amax = a(i)
        ELSEIF(a(i) < amin) THEN
            amin = a(i)
        END IF
    END DO
    !データの合計の計算
```

```

s = 0.0
DO i=1,n
    s = s + a(i)
END DO
!平均はデータの総和をデータの個数で割ればよい.
amean = s / FLOAT(n) !nは整数型なので, 実数型に変換して割るようにする.
!標準偏差を計算するために, まず平均との差の2乗和の計算. 変数sを再利用する.
s=0.0
DO i=1,n
    s = s + (a(i) - amean)*(a(i) - amean) !平均との差の2乗和
END DO
!標準偏差の計算
asigma = sqrt( s/float(n) ) !平均との差の2乗和の平均の平方根だからこうなる
END IF
RETURN
END SUBROUTINE VALUES

```

結果は次のよう出力される.

点の総数は?

4

点 1 の x, y, z 座標を入力してください.

1 0 0

点 2 の x, y, z 座標を入力してください.

1 1 0

点 3 の x, y, z 座標を入力してください.

1 1 1

点 4 の x, y, z 座標を入力してください.

2 2 2

X	Y	Z	DISTANCE
0.10000000E+01	0.00000000E+00	0.00000000E+00	0.10000000E+01
0.10000000E+01	0.10000000E+01	0.00000000E+00	0.14142135E+01
0.10000000E+01	0.10000000E+01	0.10000000E+01	0.17320508E+01
0.20000000E+01	0.20000000E+01	0.20000000E+01	0.34641016E+01

原点からの距離の最大値 : 0.34641016E+01
原点からの距離の最小値 : 0.10000000E+01
原点からの距離の平均値 : 0.19025915E+01
原点からの距離の標準偏差 : 0.93816072E+00