

情報処理概論

第6回 配列 1

情報基盤研究開発センター 谷本 輝夫

今日の予習

```
program average
  implicit none
  integer, parameter :: number = 5
  integer, dimension(number) :: score
  integer :: total, i
  real(8) :: ave
  intrinsic dble

  ! Input data
  do i = 1, number
    write(*, *) 'No. ', i, ':'
    read(*, *) score(i)
  end do
  ! Calculate average
  total = 0
  do i = 1, number
    total = total + score(i)
  end do
  ave = dble(total) / dble(number)
  ! Print results
  write(*, *) 'Average = ', ave
  do i = 1, number
    write(*, *) 'No. ', i, ' : ', score(i)-ave
  end do
stop
end program
```

今日の内容

- ▶ 配列
- ▶ 定数 (パラメータ)

プログラムの内容

- ▶ 5人分のテストの点数を入力して平均点を計算し、さらにそれぞれの点数と平均点の差を算出する
- ▶ 例えば5人の点数が以下の場合：
50点, 55点, 60点, 60点, 75点

平均 : 61点

平均との差 : -11点, -6点, -1点, -1点, 14点

実行イメージ

No. 1:

20

No. 2:

35

No. 3:

73

No. 4:

42

No. 5:

58

Average = 45.6

No. 1: -25.6

No. 2: -10.6

No. 3: 27.4

No. 4: -3.6

No. 5: 12.4

プログラム例 1

```
program average
  implicit none
  integer :: a, b, c, d, e
  real(8) :: ave
  intrinsic dble

  write(*, *) 'No. 1:'
  read(*, *) a
  write(*, *) 'No. 2:'
  read(*, *) b
  write(*, *) 'No. 3:'
  read(*, *) c
  write(*, *) 'No. 4:'
  read(*, *) d
  write(*, *) 'No. 5:'
  read(*, *) e

  ave = dble(a + b + c + d + e) / 5.0d0

  write(*, *) 'Average = ', ave
  write(*, *) '  No. 1: ', dble(a) - ave
  write(*, *) '  No. 2: ', dble(b) - ave
  write(*, *) '  No. 3: ', dble(c) - ave
  write(*, *) '  No. 4: ', dble(d) - ave
  write(*, *) '  No. 5: ', dble(e) - ave
  stop
end program
```

このプログラムの問題点

- ▶ 人数が変わると書き換えが大変
- ▶ 例えば 100人になったら？
→ 作業量が非現実的
- ▶ 以下のようにベクトル (= 配列) の添え字を使って計算できると便利

$$ave = \frac{\sum_{i=1}^{100} a_i}{100}$$

”配列変数”を使えばできる

配列変数

- ▶ 数学のベクトルや行列に似た概念

配列変数

a(1:5)

24	55	36	75	88
----	----	----	----	----

一般の変数

a

24

 b

55

- ▶ 使用例)
 - ▶ 名前と添え字（位置を示す番号）で値を格納、参照

```
a(3) = 24
```

```
do i = 1, 10  
  b(i) = a(i) + b(i)  
end do
```


配列変数の宣言

型, dimension(範囲) :: 配列変数名

- ▶ dimension で範囲を指定
- ▶ 範囲： 開始番号：終了番号
 - ▶ 開始番号を省略した場合、1が指定されたとみなす
- ▶ 例
 - ▶ score(0) ~ score(10) を利用

```
integer, dimension(0:10) :: score
```

- ▶ a(1) ~ a(100) を利用

```
integer, dimension(100) :: a
```

→ dimension(1:100) と同じ


配列変数の代入, 参照

- ▶ 位置は整数で指定：
 - ▶ 整数変数や整数式も利用可能
 - ▶ 例)

$$a(i) = b(i * 2 + j)$$

- ▶ 実数の計算でも位置は整数
 - ▶ 間違いの例)

$$~~a(1.000) = 2.000~~$$

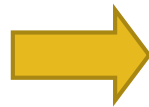

$$a(1) = 2.000$$

配列変数と繰り返し

- ▶ 配列変数は繰り返し（do文）の中で用いると便利

データの入力

```
write(*, *) 'No. 1:'  
read(*, *) a  
...  
write(*, *) 'No. 5:'  
read(*, *) e
```



```
do i = 1, 5  
  write(*, *) 'No. ', i, ':'  
  read(*, *) score(i)  
end do
```

平均点の計算

```
ave = dble(a+b+c+d+e)/5D0
```



```
total = 0D0  
do i = 1, 5  
  total = total + score(i)  
end do  
ave = dble(total) / 5D0
```

今日の予習 (再掲)

```
program average
  implicit none
  integer, parameter :: number = 5
  integer, dimension(number) :: score
  integer :: total, i
  real(8) :: ave
  intrinsic dble

  ! Input data
  do i = 1, number
    write(*, *) 'No. ', i, ':'
    read(*, *) score(i)
  end do
  ! Calculate average
  total = 0
  do i = 1, number
    total = total + score(i)
  end do
  ave = dble(total) / dble(number)
  ! Print results
  write(*, *) 'Average = ', ave
  do i = 1, number
    write(*, *) 'No. ', i, ' : ', score(i)-ave
  end do
stop
end program
```

今日の内容

- ▶ 配列
- ▶ 定数 (パラメータ)

定数（パラメータ）の利用

- ▶ プログラム中で何度も出てくる数値に名前を付ける
 - ▶ 例) 前ページのプログラム中の 5 (及び 5D0) に number という名前を付ける

```
integer, dimension(5) :: a
```

```
do i = 1, 5
```

```
do i = 1, 5
```

```
ave = dble(total) / 5d0
```

```
do i = 1, 5
```



```
integer, parameter :: number = 5
```

```
integer, dimension(number) :: a
```

```
do i = 1, number
```

```
do i = 1, number
```

```
ave = dble(total) / dble(number)
```

```
do i = 1, number
```

なぜ、数値の代わりに定数を使うか？

定数の利用

- ▶ 数値を一箇所で管理できる
= 変更が一か所で済む
- ▶ 数値の意味が分かりやすい

```
do i = 1, 5
```



```
do i = 1, number
```

- ▶ 宣言

```
型, parameter :: 定数名 = 値
```

定数と変数の違い

- ▶ 定数は値を変更できない
- ▶ 配列変数の範囲指定にも利用可能
 - ▶ 間違いの例

```
integer :: number = 5  
real(8), dimension(number) :: score
```

- ▶ 正しい例

```
integer, parameter :: number = 5  
real(8), dimension(number) :: score
```


演習：数値の正規化

- ▶ 正の実数10個を入力し、それらを以下の通り正規化する
 - ▶ 10個のうちの最大の実数に対する比率によって全実数を0.0~1.0の値に変換する。
 - ▶ 例)
 - ▶ 入力データ
20 40 14 50 100 22 43 0.4 49 2.5
 - ▶ 出力
0.2 0.4 0.14 0.5 1.0 0.22 0.43 0.04 0.49 0.025
- ▶ 注意点：適切に配列と定数を使うこと

実行イメージ

```
No. 1 :  
10  
No. 2 :  
20  
No. 3 :  
30  
No. 4 :  
40  
No. 5 :  
50  
No. 6 :  
60  
No. 7 :  
70  
No. 8 :  
80  
No. 9 :  
90  
No. 10 :  
100  
Normalized data 1 : 0.1000000000000000  
Normalized data 2 : 0.2000000000000000  
Normalized data 3 : 0.3000000000000000  
Normalized data 4 : 0.4000000000000000  
Normalized data 5 : 0.5000000000000000  
Normalized data 6 : 0.6000000000000000  
Normalized data 7 : 0.7000000000000000  
Normalized data 8 : 0.8000000000000000  
Normalized data 9 : 0.9000000000000000  
Normalized data 10 : 1.0000000000000000
```