

# 情報処理概論

第9回 ここまでのおさらい

情報基盤研究開発センター 谷本 輝夫

# 今回の内容

- ▶ 前回の課題 解答例
- ▶ 横に続けて表示する方法

# 前回の演習

- ▶ 「n人の身長と体重をファイルから読み込み、BMIを計算してファイルに書き出す」
  - ▶ 人数nもファイルから読み込む
  - ▶ BMIの計算式は、

$$BMI = \text{体重}[\text{kg}] \div (\text{身長}[\text{m}])^2$$

- ▶ 時間に余裕がある人は、以下にも挑戦
  - ▶ 「BMIが普通体重の範囲外であるものについては、マークを併せて表示する」
  - ▶ 但し、日本人の適正範囲（普通体重）は、18.5以上、25未満

# BMIプログラム 解答例

```
program Medical_checkup
  implicit none
  real(8) :: height, weight, BMI
  integer :: number, i
  character :: mark

  open(10, file="list.dat")
  open(11, file="BMI.dat")

  read(10, *) number
  do i = 1, number
    mark = ""
    read(10, *) height, weight
    BMI = weight / ((height/100) ** 2)
    if (BMI < 18.5 .OR. 25 <= BMI) then
      mark = "*"
    endif
    write(11, '(f5.1,2X,f5.2,2X,A,F4.1)') height, weight, mark, BMI
  end do

  close(10)
  close(11)
stop
end program
```

# 今回の内容

- ▶ 前回の課題 解答例
- ▶ 横に続けて表示する方法

# “改行しない”表示の利用法 (1/4)

- ▶ 例えば、以下のような表を表示したい

$1*1=$	1	$1*2=$	2	$1*3=$	3	$1*4=$	4	$1*5=$	5	$1*6=$	6	$1*7=$	7	$1*8=$	8	$1*9=$	9
$2*1=$	2	$2*2=$	4	$2*3=$	6	$2*4=$	8	$2*5=$	10	$2*6=$	12	$2*7=$	14	$2*8=$	16	$2*9=$	18
$3*1=$	3	$3*2=$	6	$3*3=$	9	$3*4=$	12	$3*5=$	15	$3*6=$	18	$3*7=$	21	$3*8=$	24	$3*9=$	27
$4*1=$	4	$4*2=$	8	$4*3=$	12	$4*4=$	16	$4*5=$	20	$4*6=$	24	$4*7=$	28	$4*8=$	32	$4*9=$	36
$5*1=$	5	$5*2=$	10	$5*3=$	15	$5*4=$	20	$5*5=$	25	$5*6=$	30	$5*7=$	35	$5*8=$	40	$5*9=$	45
$6*1=$	6	$6*2=$	12	$6*3=$	18	$6*4=$	24	$6*5=$	30	$6*6=$	36	$6*7=$	42	$6*8=$	48	$6*9=$	54
$7*1=$	7	$7*2=$	14	$7*3=$	21	$7*4=$	28	$7*5=$	35	$7*6=$	42	$7*7=$	49	$7*8=$	56	$7*9=$	63
$8*1=$	8	$8*2=$	16	$8*3=$	24	$8*4=$	32	$8*5=$	40	$8*6=$	48	$8*7=$	56	$8*8=$	64	$8*9=$	72
$9*1=$	9	$9*2=$	18	$9*3=$	27	$9*4=$	36	$9*5=$	45	$9*6=$	54	$9*7=$	63	$9*8=$	72	$9*9=$	81

## “改行しない”表示の利用法 (2/4)

- ▶ 以下のプログラムでは、表にならない。

```
program sample1_1
implicit none
integer :: i, j

  do i = 1, 9
    do j = 1, 9
      write(*, '(i1,a,i1,a,i2,1x)')&
        i, '*', j, '=', i * j
    end do
  end do
stop
end program
```

# “改行しない”表示の利用法 (3/4)

- ▶ 以下のプログラムでもいいが...

```
program sample1_2
implicit none
integer :: i

do i = 1, 9
  write(*, '9(i1,a,i1,a,i2,1x)') &
    i, '* 1 =', i * 1, &
    i, '* 2 =', i * 2, &
    i, '* 3 =', i * 3, &
    i, '* 4 =', i * 4, &
    i, '* 5 =', i * 5, &
    i, '* 6 =', i * 6, &
    i, '* 7 =', i * 7, &
    i, '* 8 =', i * 8, &
    i, '* 9 =', i * 9
end do
stop
end program
```




# “改行しない”表示の利用法 (4/4)

- ▶ 書式指定付きの write文に

`advance='NO'`  
を追加

```
program sample1_3
implicit none
integer :: i, j

do i = 1, 9
  do j = 1, 9
    write(*, '(i1,a,i1,a,i2,1x)', advance='NO') &
      i, '*', j, '=', i * j
  end do
  write(*, *)
end do
stop
end program
```



1*1=	1	1*2=	2	1*3=	3	1*4=	4	1*5=	5	1*6=	6	1*7=	7	1*8=	8	1*9=	9
2*1=	2	2*2=	4	2*3=	6	2*4=	8	2*5=	10	2*6=	12	2*7=	14	2*8=	16	2*9=	18
3*1=	3	3*2=	6	3*3=	9	3*4=	12	3*5=	15	3*6=	18	3*7=	21	3*8=	24	3*9=	27
4*1=	4	4*2=	8	4*3=	12	4*4=	16	4*5=	20	4*6=	24	4*7=	28	4*8=	32	4*9=	36
5*1=	5	5*2=	10	5*3=	15	5*4=	20	5*5=	25	5*6=	30	5*7=	35	5*8=	40	5*9=	45
6*1=	6	6*2=	12	6*3=	18	6*4=	24	6*5=	30	6*6=	36	6*7=	42	6*8=	48	6*9=	54
7*1=	7	7*2=	14	7*3=	21	7*4=	28	7*5=	35	7*6=	42	7*7=	49	7*8=	56	7*9=	63
8*1=	8	8*2=	16	8*3=	24	8*4=	32	8*5=	40	8*6=	48	8*7=	56	8*8=	64	8*9=	72
9*1=	9	9*2=	18	9*3=	27	9*4=	36	9*5=	45	9*6=	54	9*7=	63	9*8=	72	9*9=	81

# advance='no' の利用法

- ▶ write文の書式の後に追加するだけ

**write(\*, '(書式)', advance='NO') 表示内容**

- ▶ 書式は省略できない
  - ▶ 省略した場合に表示されるエラー：

```
sample.f90(8): error #6568: ADVANCE、SIZE、または EOR  
指定子のこの使用は無効です。
```

```
write(*, *, advance='NO') &
```

^

```
コンパイルは sample.f90 で異常終了しました (コード 1)。
```

# 演習問題

- ▶ 前々回の合計点の計算プログラムがベース
- ▶ 人数、科目数、点数のデータはファイルから読み込む
  - ▶ 最初の行に人数と科目数
  - ▶ 次の行からは、一行に一人の点数を科目数分
- ▶ 人数と科目数をファイルから読み込むので、当然割り付け配列を使って
- ▶ 結果は、一目で分かる表の形式で画面に出力
  - ▶ 個人の合計点と平均点、科目ごとの合計点と平均点
- ▶ 完成した人は、
  1. 偏差値を計算し、表示する
  2. 科目ごとの最高得点にマークを付ける
  3. 総得点の順位を表示する
  - ▶ など...