

## 2019年度前期 金曜4限 情報処理演習

第3回  
2019/05/10  
平山 修久



名古屋大学減災連携研究センター  
Disaster Mitigation Research Center, NAGOYA UNIVERSITY

## 前回までの復習

### > Step 1 プログラムを書く

- ー エディタ（メモ帳）でソースコードを入力し、ファイル「**\*.f90**」（拡張子）で保存。

### > Step 2 プログラムをコンパイルする

- ー コマンドプロンプトで、「**gfortran プログラム名**」で、「**a.exe**」の実行ファイルを生成。

### > Step 3 プログラムを実行

- ー コマンドプロンプトで、「**a.exe**」と入力してEnter

## コマンドプロンプトの基本

「>」マークの左側は現在のディレクトリ（カレントディレクトリ）。  
基本的なコマンド：

**cd** 書式：cd パス名

パス名を省略すると、カレントディレクトリのパスを出力。Zドライブの直下にある「fortran」ディレクトリに移動したら、「z:\fortran>」となる。

**dir** 書式：dir パス名

指定したパスのディレクトリやファイル一覧を表示。パスを省略すればカレントディレクトリの情報を表示

**cls**

コマンドプロンプトの表示をクリアする。

**exit**

コマンドプロンプトを終了する。

「↑」「↓」

入力履歴の表示

## 実際にプログラムを動かしてみよう1/4

- ＞ エディタを開く。スタート→プログラム→アクセサリ→メモ帳、もしくはWindows+R→notepad.exeを入力
- ＞ 配布した資料のプログラム例を入力。入力は**全て半角**。プログラムを見やすくするため、字下げするとよい。
- ＞ 書いたプログラムを、「sample0412.f90」という名前のファイルとして、fortranディレクトリ内に保存する。
- ＞ ファイル名が「sample0412.f90.txt」となった場合には、「.txt」を削除。
- ＞ ファイルが多くなったときは、fortranディレクトリ内に「フォルダ」（名前は日付の4桁数字「0510」など）を作成するとよい。



# fortranディレクトリ内にフォルダを作成

Windowsのエクスプローラーでfortranディレクトリを表示し、右クリックして「新規フォルダを作成」でフォルダを作成する。

コマンドプロンプトで、

**mkdir** 書式：mkdir ディレクトリ名

カレントディレクトリに「ディレクトリ名」の新しいディレクトリを作成する。



## 実際にプログラムを動かしてみよう2/4

- ＞ コマンドプロンプトを起動。デスクトップ上のショートカットをクリック, スタート→プログラム→アクセサリ→コマンドプロンプト, Windows+R→cmdと入力
- ＞ プログラムファイルを保存したディレクトリ「fortran」に移動
  - cd ¥fortran**
- ＞ コンパイラgfortranを使ってコンパイルする。
  - gfortran sample0412.f90**
- ＞ コンパイラに成功するとプログラムファイルが存在するディレクトリ内に**a.exe**という実行ファイルが作成される。



## 実際にプログラムを動かしてみよう3/4

- ＞ 実行ファイルを実行する。コマンドプロンプト上でa.exeと入力してEnter

```
a.exe
```



## ソースコードを修正する4/4

- ＞ .f90の拡張子のファイルを開く
  - ー 「.f90」の拡張子のファイルをダブルクリック
  - ー ウィンドウが出現したら、**一覧からプログラムを選択する**にチェックを入れてOKをクリック
  - ー 「**NotePad (メモ帳)**」をクリック
  - ー **この種類のファイルを開くときは、選択したプログラムをいつも使う**にチェックを入れてOKをクリック



# Windowsでスクリーンショットを撮る

- ＞ キーボードの「PrintScreen」, 「Alt + PrintScreen」 キーを利用する。
- ＞ PrintScreenでデスクトップ全体
- ＞ Alt + PrintScreenで最前面のウィンドウのみ
- ＞ その後, ペイント (「スタート」 → 「プログラム」 → 「アクセサリ」 → 「ペイント」) を起動し, 貼り付け (Control + V, もしくは右クリックで貼り付け)
- ＞ ファイルを保存



## 変数

- ＞ Fortran変数の型は, その変数に代入される値の型を決定する。ゆえに, 型宣言文によって, Fortranプログラムの各変数の型を宣言する。

INTEGER :: リスト

REAL :: リスト

```
INTEGER :: NumValues, Sum
```

```
REAL :: Mass, Velocity
```

CHARACTER(n) :: リスト

```
CHARACTER(15) :: FirstName, LastName
```

```
CHARACTER :: Initial
```

```
CHARACTER(15) :: Name, L_Name*20, Initial*1
```

## IMPLICIT NONE文

- ＞ Fortranでは、型宣言文で明示的に型を宣言しない変数は、**暗黙の型宣言**に従って型を割り当てられる。名前がI, J, K, L, M, Nおよび対応する小文字で始まる未宣言の識別子は整数型 (INTEGER) となり、それ以外はすべて実数型 (REAL)
- ＞ この暗黙の型宣言を無効にする文

```
IMPLICIT NONE
```

## 変数の初期化

- ＞ Fortranではすべての変数が初期状態では未定義

```
REAL :: W = 1.0, X = 2.5, Y = -7.73
```

- ＞ 名前付定数：PARAMETER属性

型指定子, PARAMETER :: リスト

```
INTEGER, PARAMETER :: Limit = 50
```

```
REAL, PARAMETER :: Pi = 3.141593, TwoPi =  
2.0 * Pi
```

```
CHARACTER(*), PARAMETER :: Units = "cm"
```



## 数値関数

ABS(x) xの絶対値  
COS(x) xラジアンのコサイン（余弦）  
EXP(x) 指数関数  
INT(x) xの整数部分  
FLOOR(x) x以下の最大の整数  
FRACTION(x) xの小数部分  
LOG(x) xの自然対数  
MAX(x1, x2, ..., xn) x1,,,xnの最大値  
MIN(x1, x2, ..., xn) x1,,,xnの最小値  
MOD(x, y) xをyで割った余り,  $x - \text{INT}(x/y)*y$   
NINT(x) x1の最も近い整数  
REAL(x) xを実数型に変換  
SIN(x) xラジアンのサイン（正弦）  
SQRT(x) x1の平方根  
TAN(x) xラジアンのタンジェント（正接）



## 代入文

### > 変数=式

```
REAL :: XCoordinate, YCoordinate, Alpha, Beta
INTEGER :: Number, Term, N
XCoordinate = 5.23
YCoordinate = SQRT(25.0)
Number = 17
Term = Number/3+2
XCoordinate = 2.0*XCoordinate
N=9
Alpha = 3
Beta = (N+3)/5
I = 3.14159
Kappa = X/2.0
Mu = 1.0/X
```



## 入出力

### > PRINT \*, 出力並び

```
PRINT *, "At time", Time, "seconds"  
PRINT *
```

### > READ \*, 入力並び

```
READ *, InitialHeight, InitialVelocity, Time  
100.0, 90.0, 4.5  
100.0 90.0 4.5  
100.0 90.0  
4.5
```



## プログラムの構成と書式

### > Fortranプログラムの一般的な構成

- PROGRAM 名前
- 宣言部
- 実行部
- 副プログラム部
- END PROGRAM 名前



## 単純論理式

### > 論理定数（.TRUE.と.FALSE.）

式1 関係演算子 式2

演算子	意味
< または .LT.	より小さい
> または .GT.	より大きい
.EQ. または ==	等しい
<= または .LE.	以下
>= または .GE.	以上
/= または .NE.	等しくない

## 複合論理式

### > .NOT.（否定），.AND.（論理積），.OR.（論理和），.EQV.（等価），.NEQV.（排他的論理和）

p	.NOT. p
.TRUE.	.FALSE.
.FALSE.	.TRUE.

p	q	p .AND. q	p .OR. q	p .EQV. q	p .NEQV. q
.TRUE.	.TRUE.	.TRUE.	.TRUE.	.FALSE.	.TRUE.
.TRUE.	.FALSE.	.FALSE.	.TRUE.	.FALSE.	.TRUE.
.FALSE.	.TRUE.	.FALSE.	.TRUE.	.FALSE.	.TURE.
.FALSE.	.FALSE.	.FALSE.	.FALSE.	.TRUE.	.FALSE.

# IF構文

## ＞ 最も単純な選択構造

```
IF (論理式) THEN
```

```
    文の並び
```

```
END IF
```

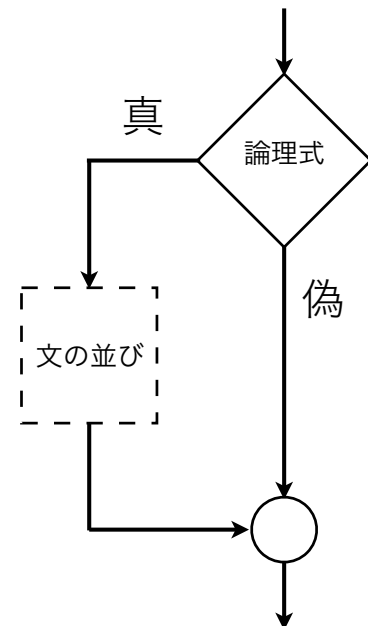
```
IF (X >= 0) THEN
```

```
    Y = X * X
```

```
    Z = Sqrt(X)
```

```
END IF
```

```
IF (1.5 <= X .AND. x <= 2.5) PRINT *,X
```



# IF構文の汎用形式

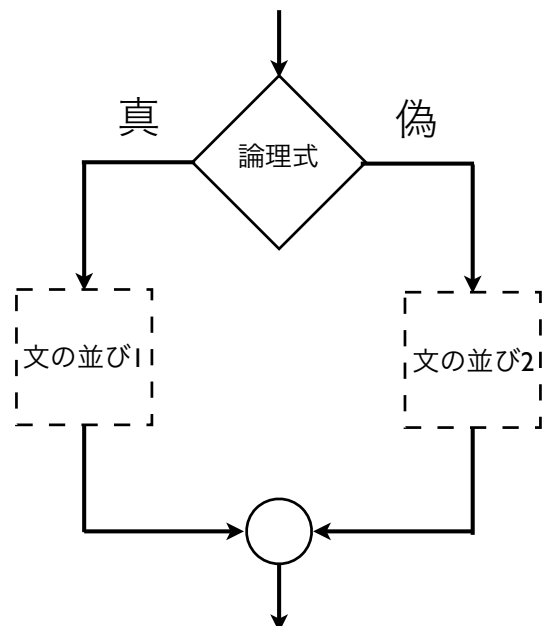
```
IF (論理式) THEN
```

```
    文の並び1
```

```
ELSE
```

```
    文の並び2
```

```
END IF
```



## 課題3\_1 二次方程式

$$A x^2 + B x + C = 0$$

＞ 解の公式を使って解く

$$\frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

＞ アルゴリズム

- 1.A, B, Cを入力する。
- 2.判別式 (Discriminant) を計算する。
- 3.Discriminant  $\geq 0$ であれば, 2つの根を計算して表示
- 4.そうでなければ判別式の値と実根なしのメッセージを表示

## IF-ELSE IF構文

＞ 3つ以上の選択肢が含まれるとき

```
IF (論理式1) THEN
    文の並び1
ELSE
    IF (論理式2) THEN
        文の並び2
    ELSE
        文の並び3
    END IF
END IF
```

＞ IF-ELSE IF構文を使うと

```
IF (論理式1) THEN
    文の並び1
ELSE IF (論理式2) THEN
    文の並び2
ELSE IF (論理式3) THEN
    文の並び3
ELSE
    文の並びn
END IF
```



## IF-ELSE IF構文例

```
IF (N >= 90) THEN
    PRINT *, "S"
ELSE IF (N >= 80) THEN
    PRINT *, "A"
ELSE IF (N >= 70) THEN
    PRINT *, "B"
ELSE IF (N >= 60) THEN
    PRINT *, "C"
ELSE
    PRINT *, "F"
END IF
```



## CASE構文

```
SELECT CASE (場合式)
    CASE (場合値リスト1)
        文の並び1
    CASE (場合値リスト2)
        文の並び2
    CASE (場合値リスト3)
        文の並び3
END SELECT
```

- > 場合式は、整数式、文字式、論理式
- > 場合値リストは、場合式が取りうる1つ以上の値を括弧で囲んだリストか、キーワードDEFAULT
- > (値)【単一の値】，(値1:値2)【値1から値2までの範囲】，(値1:)【値1以上のすべての値の集合】，(:値2)【値2以下のすべての値の集合】



# CASE構文例

```
SELECT CASE (ClassCode)
  CASE (1)
    PRINT *, "Freshman"
  CASE (2)
    PRINT *, "Sophomore"
  CASE (3)
    PRINT *, "Junior"
  CASE (4)
    PRINT *, "Senior"
  CASE (5)
    PRINT *, "Graduate"
  CASE DEFAULT
    PRINT *, "Illegal class code", ClassCode
END SELECT
```



## 課題3\_2

- ＞ 名古屋市上下水道局の水道料金（一般用口径13ミリ）を計算し，使用水量と水道料金（小数点以下は切り捨て）を表示するプログラムを作成する。
- ＞ アルゴリズム
  1. 前回の指示値，今回の指示値を入力する。
  2. 今回の使用水量を計算する。
  3. 使用水量で選択実行
    - 1) 使用水量が13～20のとき，料金＝基本料金+(使用水量-12)×10.80
    - 2) 使用水量が21～40のとき，料金＝基本料金+10.80×8+(使用水量-20)×166.32
    - 3) 使用水量が40～60のとき，料金＝基本料金+10.80×8+166.32×20+(使用水量-40)×228.96
    - 4) 使用水量が12以下のとき，料金＝基本料金
  4. 使用水量と料金を表示する。



## 演習課題3

＞ 提出物：4つ

1.kadai3\_1.f90（プログラム）

2.kadai3\_1.png（実行画像ファイル）

3.kadai3\_2.f90（プログラム）

4.kadai3\_2.png（実行画像ファイル）