

2019年度前期 金曜4限 情報処理演習

第2回
2019/04/26
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

前回の復習

- > Step 1 プログラムを書く
 - 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
 - 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
 - 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に、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」となった場合は、「.txt」を削除。

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
 - 「.f90」の拡張子のファイルをダブルクリック
 - 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
 - 「**NotePad (メモ帳)**」をクリック
 - 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

データ型, 定数, 変数

> Fortranには5種類の基本データ型

INTEGER : 整数 0, 137, -2516, 17745

REAL : 実数 1.234, -0.01536, +56473.,
337.456, 3.37456E2

COMPLEX : 複素数 (a, b), (1.0, 1.0), (-6.0,
7.2)

CHARACTER : 文字列, 二重引用符またはアポスト
ロフィで囲まなければならず, 両端に同じ記号。

LOGICAL : 論理変数, .TRUE., .FALSE.

識別子

> プログラム, 定数, 変数など, プログラム中の要
素を識別するために使う名前。

> 英文字で始まり, 最大30文字までの英文字, 数
字, 下線

Mass

Rate

Velocity

Speed_of_Light

— Fortran90では大文字と小文字を区別しない。

— 一般的に, Fortranのキーワードは大文字, プログラ
マが定義する識別子は小文字 (最初も文字は大文字)

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"
```

演算と関数

> 数値演算

+ (加算) , - (減算) , * (乗算) , / (除算) ,
** (累乗)

```
B**2 - 4 * A * C
```

> 型混合式

— 整数の量と実数の量と結びつけると、整数の量が実数の量に変換され、演算結果は実数型となる。

1.0 / 4

3.0 + 8 / 5 と 3 + 8.0 / 5 との違い

演算の優先規則

- › 最初に、すべての累乗が実行される。累乗の累乗の累乗…の場合には、右から左に実行。
- › 次に、すべての乗算と除算が、左から右へ記述されている順に実行される。
- › 最後に、加算と減算が、左から右へ記述されている順に実行される。

$$2^{**}3^{**}2 = 2^{**}9 = 512$$

$$10/5*2 = 2*2 = 4$$

$$2+4^{**}2/2 = 2+16/2 = 2+8 = 10$$

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 - 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）

文字演算

> 連結演算//

```
“centi” // “meters” ⇒ “centimeters”
SquareUnit = “square ”
SquareUnit // “centi”// “meters”
⇒ “square centimeters”
```

> 部分列にアクセス：文字型変数の値や文字定数から部分列を抽出。部分列の開始文字と終了文字の位置をコロン (:) で区切り、括弧で囲んで、変数や定数の後に置く。

```
Units = “centimeters”
Units(4:7) ⇒ “time”
```

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

演習課題2

> 提出物：3つ

- 1.kadai2_1.f90 (プログラム)
- 2.kadai2_2.png (画像ファイル)
- 3.kadai2_3.f90 (プログラム)