情報処理概論(第4回)

情報基盤センター　井上 仁

授業内容(第4回)

- Fortranの文法の説明
  - 注釈行
  - PROGRAM文、STOP文、END文
  - 型宣言文
    - 整数型
  - 入出力文
  - 代入文
  - IF文
  - IF構文
    - IF THEN文、ELSE文、ELSE IF文、END IF文

例題プログラム(welcome.f90)

```
! Welcome to Fortran World

      PROGRAM welcome

      WRITE(*,*) 'Welcome to Fortran World!'

      STOP

      END PROGRAM welcome
```

例題プログラム(ex1_1a.f90)

```
! ex1_1a
! 和差積商の計算
      PROGRAM ex1_1a

      INTEGER :: a,b,wa,sa,seki,sho

      WRITE(*,*) 'Enter two integers'
      READ(*,*) a,b
      wa=a+b
      sa=a-b
      seki=a*b
      sho=a/b
      WRITE(*,*) 'a=',a,' b=',b
      WRITE(*,*) 'wa=',wa,' sa=',sa,'seki=',seki,' sho=',sho
      STOP

      END PROGRAM ex1_1a
```

例題プログラムのPascal版

```
program ex1_1a;

   var a,b,wa,sa,seki,sho: integer;

   begin
      writeln('Enter two integers');
      readln(a,b);
      wa:=a+b;
      sa:=a-b;
      seki:=a*b;
      sho:=a div b;
      writeln('a=',a,' b=',b);
      writeln('wa=',wa,' sa=',sa,' seki=',seki,' sho=',sho)
   end.
```

注釈行、PROGRAM文

- 注釈行
  - !で始まる行
- PROGRAM文
  - プログラムの開始を宣言する文
    (例) PROGRAM WELCOME
  - プログラム名
    - 英字(AからZ)、数字(0から9)、下線(_)
    - 先頭の文字は英字
    - 長さは31文字以下

END文、STOP文

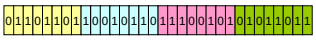
- END文
  - プログラムの終わりを表す文
    (例)END
    END PROGRAM WELCOME
  - PROGRAM文と対に用いる
    PROGRAM WELCOME
    …
    END PROGRAM WELCOME
- STOP文
  - プログラムの実行を停止する命令

定数

- 整定数
  - 123　-456　123456　※ 123,456は誤り
- 実定数
  - 3.14159　123.　1.2E3　4.5D-67
- 複素定数
  - (実数部, 虚数部)
- 論理定数
  - .TRUE.　(真)　.FALSE.　(偽)
- 文字定数
  - 'Welcome to Fortran World'
  - 'Don''t'
  - "Don't"

変数の型(1)

- プログラムの実行中に変化する値
- INTEGER, INTEGER(4)
  - 4バイトの記憶領域
  - 整数を正確に表現
  - -2,147,483,648 ～ 2,147,483,647
    -2^(31) ～ 2^31 - 1



※ INTEGER(1), INTEGER(2), INTEGER(8)

- 変数名
  - 英字(AからZ)、数字(0から9)、下線(_)
  - 先頭の文字は英字
  - 長さは31文字以下
    ※プログラム名と同じ

入出力文

- 入力文
  - READ(*,*) 変数の並び
    - READ(*,*) a
    - READ(*,*) a,b
- 出力文
  - WRITE(*,*) 値の並び
    - WRITE(*,*) 'Welcome to Fortran World'
    - WRITE(*,*) 'a=', a

代入文

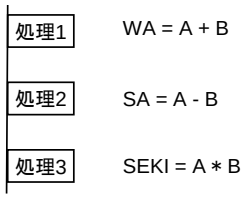
- 右辺の値を評価して、左辺の変数の値として代入
  - wa = a + b
  - pi = 3.14159
- 左辺に定数はこない
  - 3 = a + b 誤り
- 変数の型に注意
  - INTEGER :: a
  - a = 'Welcome to Fortran World' 誤り

プログラムの基本制御構造

- 連接
  - 順に処理する
- 選択
  - 条件によって処理を選択する
- 反復
  - 一定回数、あるいは条件を満たすあいだ処理を繰り返す

連接

■順に処理する



プログラムのコンパイル

■コンパイル(翻訳)とリンク(結合編集)

% ls
a.out* ex1_1a.f90 welcome.f90
% frt ex1_1a.f90 **-o** ex1_1a
% ls
a.out* ex1_1a* ex1_1a.f90 welcome.f90
※ex1_1aというファイルができている
※＊は実行可能なファイルを表す

プログラムの実行

```
% ./ex1_1a
  Enter two integers
4  2
  a= 4  b= 2
  wa= 6  sa= 2  seki= 8  sho= 2
```

※．は、カレントディレクトリを表す

例外処理

■ex1_1a.f90のプログラムで、例えば a = 4 、b=0 を入力したらどうなるか?

プログラムの再利用

■例題1-1aのプログラムをコピーする

% cp ex1_1a.f90 ex1_1b.f90

例題プログラム(ex1_1b.f90)

```
! ex1_1b
! 和差積商の計算
      PROGRAM ex1_1b

      INTEGER :: a,b,wa,sa,seki,sho

      WRITE(*,*) 'Enter two integers'
      READ(*,*) a,b
      wa=a+b
      sa=a-b
      seki=a*b
      IF (b.NE.0) sho=a/b
      WRITE(*,*) 'a=',a,' b=',b
      IF (b.NE.0) THEN
         WRITE(*,*) 'wa=',wa,' sa=',sa,' seki=',seki,' sho=',sho
      ELSE
         WRITE(*,*) 'wa=',wa,' sa=',sa,' seki=',seki
      ENDIF
      STOP

      END PROGRAM ex1_1b
```

例題プログラムのPascal版

```
program ex1_1b;

   var a,b,wa,sa,seki,sho: integer;

   begin
      writeln('Enter two integers');
      readln(a,b);
      wa:=a+b;
      sa:=a-b;
      if b<>0 then sho:=a div b;
      writeln('a=',a,' b=',b);
      if b<>0 then
         writeln('wa=',wa,' sa=',sa,' seki=',seki,' sho=',sho)
      else
         writeln('wa=',wa,' sa=',sa,' seki=',seki)
   end.
```

IF文

■IF (論理式) 実行文

◆論理式が真のとき実行文が実行される

(例)

IF (b.NE.0) sho=a/b

IF (b/=0) sho=a/b

◆関係演算子

.GT. > .GE. >= .LT. <

.LE. <= .EQ. == .NE. /=

IF構文(1)

IF (論理式) THEN

ブロック (論理式が真のとき実行)

ENDIF

(例)

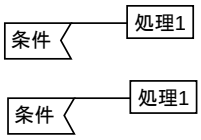
```
IF (b.NE.0) THEN
   sho=a/b
ENDIF
```

ブロックは、複数の実行文

```
IF (b.NE.0) THEN
   WRITE(*,*) 'b is not 0'
   sho = a / b
ENDIF
```

選択(1)

■条件によって処理を選択する




```
IF (B.NE.0) SHO = A / B
```



```
IF (B.NE.0) THEN
   SHO = A / B
ENDIF
```

条件によって、処理1が実行されるか、実行されない

IF構文(2)

IF (論理式) THEN

ブロック1 (論理式が真のとき実行)

ELSE

ブロック2 (論理式が偽のとき実行)

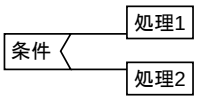
ENDIF

(例)

```
IF (b.NE.0) THEN
    WRITE(*,*) 'wa=',wa,' sa=',sa,' seki=',seki,' sho=',sho
ELSE
    WRITE(*,*) 'wa=',wa,' sa=',sa,' seki=',seki
ENDIF
```

選択(2)

■条件によって処理を選択する




```
IF (B.NE.0) THEN
   WRITE(*,*) ……
ELSE
   WRITE(*,*) ……
ENDIF
```

条件によって、処理1が実行されるか、処理2が実行される

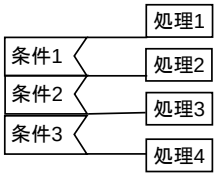
IF構文(3)

IF (論理式1) THEN
　ブロック1 (論理式1が真のとき)
ELSE IF (論理式2) THEN (論理式1が偽のとき)
　ブロック2 (論理式2が真のとき)
.....
ELSE IF (論理式n) THEN (論理式n-1が偽のとき)
　ブロックn (論理式nが真のとき)
ELSE
　ブロックn+1 (論理式nが偽のとき)
ENDIF

IF構文の例

```
IF (point >= 80) THEN
    WRITE(*,*) 'A'
ELSEIF (point >= 70) THEN
    WRITE(*,*) 'B'
ELSEIF (point >= 60) THEN
    WRITE(*,*) 'C'
ELSE
    WRITE(*,*) 'D'
ENDIF
```

選択(3)




```
IF (point >= 80) THEN
   WRITE(*,*) 'A'
ELSEIF (point >= 70) THEN
   WRITE(*,*) 'B'
ELSEIF (point >= 60) THEN
   WRITE(*,*) 'C'
ELSE
   WRITE(*,*) 'D'
ENDIF
```

条件によって、処理1が実行されるか、処理2が実行されるか、処理3が実行されるか、処理4が実行される。

演習

■各例題プログラムを入力して、実行しなさい。

次回予定(第5回)

■Fortranの文法の説明
- ◆型宣言文
  - ■実数型
- ◆ 名前付き定数
- ◆CASE構文
  - ■SELECT CASE文、CASE文、END SELECT文
- ◆DO構文
  - ■DO文、END DO文、EXIT文
- ◆CONTINUE文