

本日のテーマ

- 配布 CDROM の使い方 (前回の復習)
- Fortran のプログラムの書き方

1 前回の復習

1.1 作業フォルダの作成とソースプログラムの作成

前回作成した作業フォルダ `c:\work\T66160` にサブフォルダ→「20170417」を作成してください。次に、「メモ帳」を使って次の Fortran のプログラム (ソースプログラム) を入力して下さい。引用符「`'`」や空白文字も半角文字ですので注意してください。入力が終わったら、名前を「`menseki.f`」として、今作成した作業フォルダ `c:\work\T66160\20170417` に保存してください。保存する場所を「ローカルディスク (C:) ▶ work ▶ T66160 ▶ 20170417」にすることと、「すべてのファイル」にしてからファイル名を入力するのを忘れないようにしてください。

menseki.f

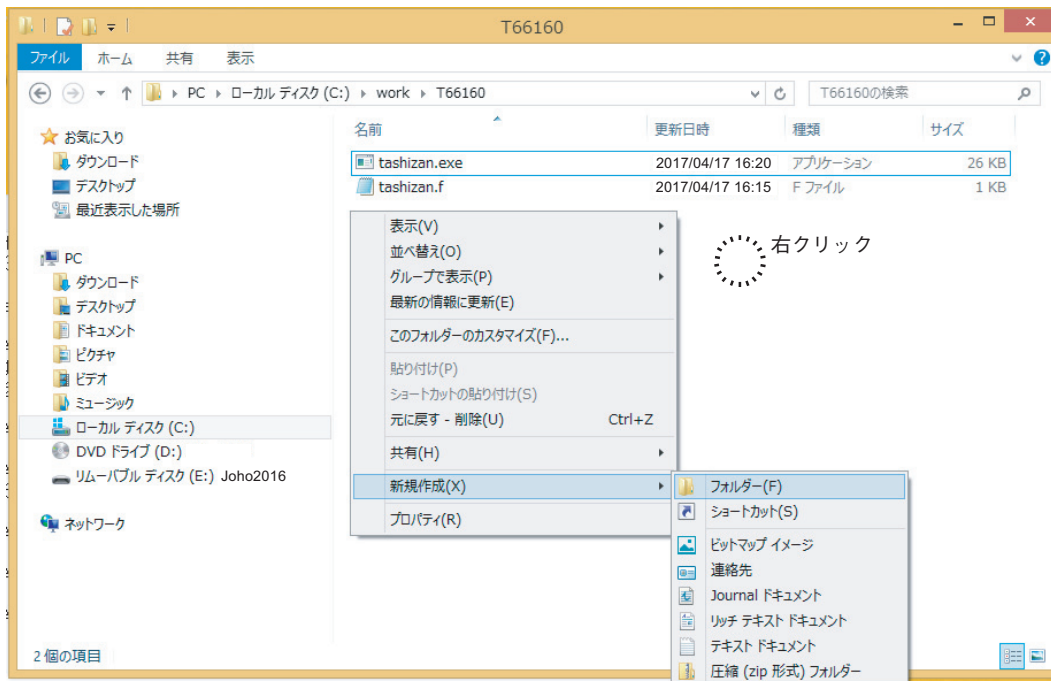
```
c23456
  program menseki
  real area, s, a, b, c

  write(6,*) ' 3辺の長さを入力して下さい. '
  read(5,*) a, b, c

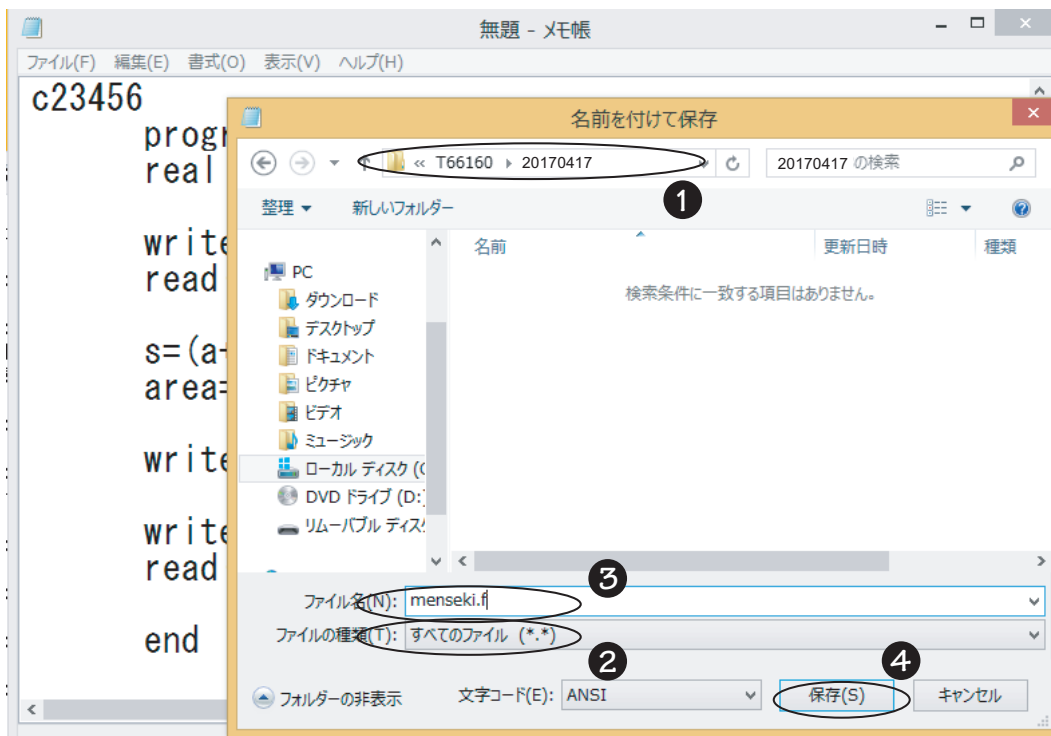
  s=(a+b+c)/2.0
  area=SQRT(s*(s-a)*(s-b)*(s-c))

  write(6,*) ' 面積は', area, ' です. '

  write(6,*) ' プログラム終了. エンターキーを押して下さい. '
  read(5,*)
  stop
end
```



作業フォルダの作成



メモ帳でソースプログラムを作成し保存

1.2 ソースプログラムをコンパイル，実行する

次に作成したソースプログラム `menseki.f` をコンパイルするために，コマンドプロンプトを起動します。コマンドプロンプトが起動したら，前回の資料と下の入力例を参考に，コンパイルと実行を行ってください。

コマンドプロンプトでのキー入力と出力の様子

```
Microsoft Windows [Version 6.0.6002]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

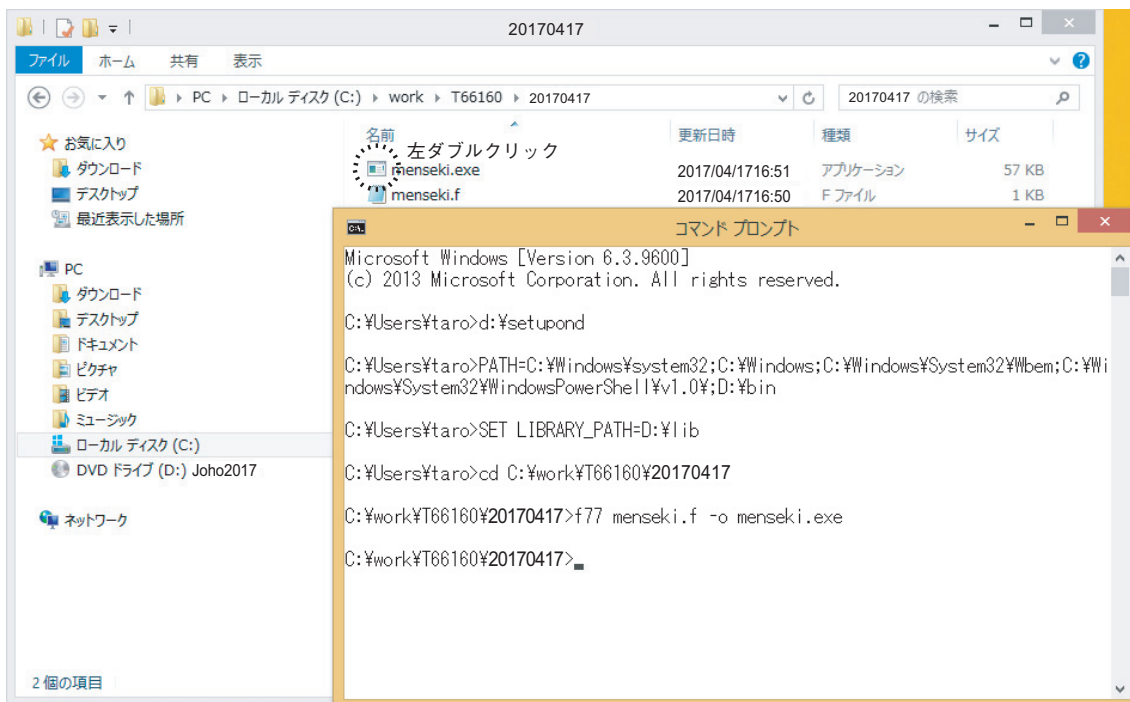
C:¥Users¥kiban>E:¥setUpOnE          ← CDROM用環境設定 (E:がCDのとき)
C:¥Users¥kiban>PATH=E:¥bin
C:¥Users¥kiban>SET LIBRARY_PATH=E:¥lib
C:¥Users¥kiban>cd ¥work¥T66160¥20170417      ←作業フォルダに移動
C:¥work¥T66160¥20170417>dir                ←フォルダ中のファイルを確認
ドライブ C のボリューム ラベルがありません。
ボリューム シリアル番号は 6C95-2D53 です

C:¥work¥T66160¥20170417 のディレクトリ
2017/04/17  16:36    <DIR>          .
2017/04/17  16:36    <DIR>          ..
2017/04/17  16:40                340 menseki.f  ← menseki.f の保存の確認
                1 個のファイル                340 バイト
                2 個のディレクトリ 104,296,501,248 バイトの空き領域

C:¥work¥T66160¥20170417>f77 menseki.f -o menseki ← menseki.f のコンパイル
                                                    ←ここで文句を言われなければコンパイル成功
C:¥work¥T66160¥20170417>menseki.exe ←生成された実行ファイル menseki.exe を実行
3 辺の長さを入力してください。                (.exe は付けなくても良い)
2.0, 2.0, 3.0                                  ←三角形の3辺の長さを適当に入れてみる。
面積は 1.98431349 です。                        ←計算結果
プログラム終了. エンターキーを押して下さい. ←エンターキーを押してプログラム終了
C:¥work¥T66160¥20170417>
```

1.3 コンパイルによって作成した実行ファイルを実行する

コンパイラでコンパイルして作成した実行ファイルを実行するには、上の例のようにコマンドプロンプト内でファイル名 menseki(.exe) を直接入力する方法の他に、コンパイルによって生成された menseki.exe を Explorer 上で左ダブルクリックする方法もあります (下図)。



1.4 .f ファイルをダブルクリックしてメモ帳を立ち上げる設定

Explorer で menseki.f を左ダブルクリックするとメモ帳が立ち上がるようにしておくとは便利です。これには拡張子の関連付け機能を使います。Explorer で menseki.f (.f ファイルなら何でも構いません) をダブルクリックをすると下図のようなウィンドウが現れるので、プログラムの一覧が表示されるので「メモ帳」を選択します。「メモ帳」が表示されないときは「別のアプリ…」をクリックして「メモ帳」を探してください。この操作を一度しておくで、以降は拡張子.f のファイルがダブルクリックされたら「メモ帳で開ける」と Windows が憶えていてくれます。



1.5 実行結果をコピー・ペーストする

実行結果をコピーしてワープロなどに貼り込むことができます。コマンドプロンプトの左上をクリックしてメニューを出し、「編集」→「範囲指定」を選んでからマウスで範囲を指定し、再び、左上をクリックしてメニューを出して、「編集」→「コピー」。次に貼り付けをしたいところで、「編集」→「貼り付け」。

2 Fortran のソースプログラム (ソースファイル) の書き方

2.1 カラムの使い方

ソースプログラムは Fortran 規格に従って正しい文法で書かなければなりません。Fortran 77 では、カラム (左端から数えた文字の位置) について次のような決まりがあります (CDROM の教科書 (¥text¥main.pdf) の p.27 参照)。

- 1 カラム目 : c または * (アスタリスク) と書いてあるとその行は単なるメモと見なされる (このような行をコメント行という)。
- 1~5 カラム目 : 行番号 (statement label) 指定用
- 6 カラム目 : ここに何か書いてあると継続行 (前の行の続き) と見なされる。
- 7~72 カラム目 : 通常の命令用 (73 カラム以降は書いても無視されるので注意)

はじめの6カラムまで上記のような特殊な意味があるので、**通常の命令は7カラム目以降**に書くてはいけません。メモ帳ではカラム位置がわからないため、サンプルプログラムの一行目では

と打ってあります。

1行打ち込み終わって次の行に書くときには、必ず Enter キーで次の行に移ってください(スペースキーをたくさん押すというのはだめ)。

2.2 program文とend文

プログラムは program 文で始まり、end 文で終わります。program 文にはプログラムの名前を付けておきます(特別な機能があるわけでもなく、program 文全体を省略することもできます)。

```
program menseki
```

この名前には、英数字、\$, _ などの一部の記号を使うことができます(一文字目に数字は使えません)。この名前と同じ名前に .f を付けた名前をファイル名とするのが良いと思います。

2.3 宣言文と実行文

program 文と end 文の間に、計算したい内容に対応した命令(実行文)を順序正しく、書いていきます。プログラム例では、次のような実行文が使われています(行の右に説明がついている文が実行文です)。

menseki.f

```
1:c23456
2:    program menseki
3:    real area, s, a, b, c
4:
5:    write(6,*)' 3辺の長さを入力して下さい. ' ! 「3辺の…」を画面表示を行う実行文
6:    read(5,*)a, b, c          ! ユーザが打ち込んだ数字をプログラムに取り込む実行文
7:
8:    s=(a+b+c)/2.0            ! ユーザが打ち込んだ3つの値から s を計算する実行文
9:    area=SQRT(s*(s-a)*(s-b)*(s-c)) ! 面積を計算する実行文, SQRT は
10:    ! 平方根(SQuare RooT)を計算する組込関数(小文字でもOK・教科書 p.50-参照)
11:    write(6,*)' 面積は', area, 'です. ' ! 計算結果 area を表示する実行文
12:
13:    write(6,*)' プログラム終了. エンターキーを押して下さい. ' ! これも実行文
14:    read(5,*)                ! これも実行文
15:    stop                      ! プログラム停止の実行文(省略可能)
16:    end                       ! これも実行文に分類されています。
```

実行文は

- 上から順に実行されていく。
- 変数を的確に使って、計算の手順がわかりやすくなるように書いていく。

のが基本です。

サンプルプログラムで使われている宣言文は、real(型宣言文)と前述の program 文です。型宣言文 real は計算に使われる変数(area, s, a, b, c)が実数(小数点のつく数値)を記憶するための変数であることを宣言しています。

2.4 変数について

Fortranに限らず計算機では数字を入れておく変数(値を入れておく箱のようなもの)を使って計算を行います。例えば、サンプルプログラムでは、まず、変数 a , b , c に値を入力させ (5, 6 行目), その値を用いて、別の変数 s に

$$(a + b + c)/2.0$$

の値を保存しています (8 行目)。さらに、この s の値と一つ前の行で読み込んだ a , b , c の値から

$$\sqrt{s(s-a)(s-b)(s-c)}$$

を計算し、その結果を別の変数 $area$ に保存しています (9 行目)。このように一度、変数に値を保存してしまえば、その後はプログラム中で、いつでも保存した変数の内容を計算に使うことができます。

2.5 =の意味

上の説明で注意が必要なことは、 $=$ は等しいという意味ではなく、左の変数に右側の値を代入するという意味であることです。つまり、 $=$ の右側は結果的に値が一つ決まるものならば何でもよいので、値(数字)や変数、式(演算)、関数など様々あり得ますが、 $=$ の左側は値をしまっておくことができる箱(変数・メモリー)でなければならないということです。たとえば、

- $a = 3.0$ (OK, 変数 a の値を 3.0 にする)
- $a = b$ (OK, 変数 b の値を変数 a に代入する)
- $a = a + 1$ (OK, 現在の a の値に 1 を足して、これを新たな a の値とする)
- $1 = a$ (NG, 1 は定数値であり、変数(記憶するためのメモリー)ではない)
- $a+1 = c$ (NG, $a+1$ は式であり、変数(記憶するためのメモリー)ではない)
- $a/b = 5$ (NG, 同上)

3 練習問題・宿題

1. a が 2.0, b が 7.5 のときの $a + b$ を計算するプログラムを作成せよ。
2. x の値をユーザに入力させて、 $y = 3x + 4$ を計算するプログラムを作成せよ。
3. プログラム中に 3 点の座標 (a_x, a_y) , (b_x, b_y) , (c_x, c_y) を与えて、この 3 点で作る三角形の面積を計算するプログラムを作成せよ。
4. 質問があれば出席確認用に配布した紙の下の方に書いておいて下さい。