Fortranプログラミング入門

-モジュール-

副プログラム

副プログラムとは…

- プログラムを小さいプログラムに分割して使う機能
- 単独では実行出来ない
- Fortranには関数とサブルーチンがある

プログラムにバグが出にくい!
バグが出ても，エラーになる部分が限定される!

プログラム単位

Fortranのプログラムを構成する基本的な構成要素を
プログラム単位と呼ぶ.

・主プログラム
　program文で始まる. 一つのプログラムに必ず一つ.
・外部副プログラム
　現在はモジュールで代替. 様々な主プログラムから
　　呼び出せる副プログラム.
・モジュール(Fortran90から追加)
　変数や副プログラムなどをまとめたもの.

プログラム単位

- 主プログラム(program文)
  - 内部関数
  - 内部サブルーチン

- 外部副プログラム
  - 外部関数
  - 外部サブルーチン

- モジュール ← 今回はこれ!!

モジュールとは

モジュールとは，関数，サブルーチン，共有変数をまとめて宣言する一つのプログラム単位．Fortran90で追加された目玉機能の一つ．

モジュールとは

module モジュール名

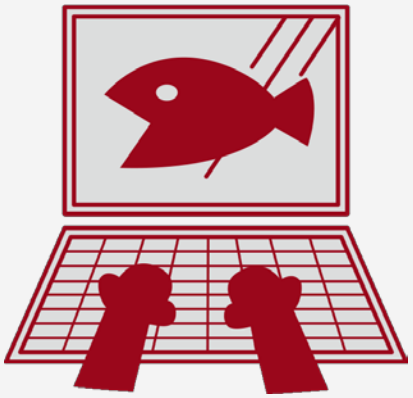
implicit none ← 暗黙の型宣言をオフに!

宣言文 ← 共有させたい変数をここに宣言!!

contains

関数/サブルーチンの宣言 ← 関数やサブルーチンの書き方は今まで通り!!

end module モジュール名



use文

☆文法 use モジュール名

・モジュールを使いたいプログラム単位の最初に書く!!

・変数の参照の制限や別名での参照方法がある(今回はなし!!)

```
module testmod
・・・
end module testmod
program examp
 use testmod
 implicit none
 ・・・・
end program examp
```

モジュールの宣言は使用するプログラムの前(別ファイルも含め)に行う

use文はimplicit noneの前に書く!!
use文で使用するモジュールを指定!

別ファイルを利用するモジュールのコンパイル

testmod.f90

```
module testmod
 implicit none
  ・・・
 contains
 ・・・
end module
```

main.f90

```
program main
 use testmod
 implicit none
 ・・・
 stop
end program main
```

コンパイルは必ず使用するモジュールが先!!

gfortran testmod.f90 main.f90

例題1

rを倍精度実数型の変数とし，適当に値を代入する．そのとき，rを半径とする球の体積を計算する関数とサブルーチンが書かれたモジュールを作成せよ

$$V = \frac{4}{3}\pi r^3$$

例題1

volmod.f90

```
module volmod
 implicit none
 contains
 function dvolfunc(r) result(V)
  real(8), intent(in) :: r
  real(8) :: pi = acos(-1d0)
  real(8) :: V
  V = 4d0/3d0*pi*r**3
 end function dvolfunc
 subroutine dvolsub(r,V)
  real(8), intent(in) :: r
  real(8) :: pi = acos(-1d0)
  real(8), intent(out) :: V
  V = 4d0/3d0*pi*r**3
 end subroutine dvolsub
end module volmod
```

main.f90

```
program vol
 use volmod
 implicit none
 real(8) :: r, V
 r = 2d0
 V = dvolfunc(r)
 write(*,*) V
 call dvolsub(r,V)
 write(*,*) V
 stop
end program vol
```

例題1

volmod.f90

```
module volmod
 implicit none
 contains
 function dvolfun
  real(8), intent(in) :: r
  real(8) :: pi = acos(-1d0)
  real(8) :: V
  V = 4d0/3d0*pi*r**3
 end function dvolfunc
 subroutine dvolsub(r,V)
  real(8), intent(in) :: r
  real(8) :: pi = acos(-1d0)
  real(8), intent(out) :: V
  V = 4d0/3d0*pi*r**3
 end subroutine dvolsub
end module volmod
```

piについていろいろ検討してみる!

main.f90

```
program vol
 use volmod
 implicit none
 r = 2d0
 V = dvolfunc(r)
 write(*,*) V
 call dvolsub(r,V)
 write(*,*) V
 stop
end program vol
```

例題1

volmod.f90

```
module volmod
 implicit none
 contains
 function dvolfun
  real(8), intent(in) :: r
  real(8) :: pi = acos(-1d0)
  real(8) :: V
  V = 4d0/3d0*pi*r**3
 end function dvolfunc
 subroutine dvolsub(r,V)
  real(8), intent(in) :: r
  real(8) :: pi = acos(-1d0)
  real(8), intent(out) :: V
  V = 4d0/3d0*pi*r**3
 end subroutine dvolsub
end module volmod
```

piについていろいろ検討してみる!

main.f90

```
program vol
 use volmod
 implicit none
 r = 2d0
 V = dvolfunc(r)
 write(*,*) V
 call dvolsub(r,V)
 write(*,*) V
 write(*,*) pi
 stop
end program vol
```

piは参照できない…

Error…

例題1

volmod.f90

```
module volmod
 implicit none
 real(8) :: pi = acos(-1d0)
 contains
 function dvolfunc(r) result(V)
  real(8), intent(in) :: r
  real(8) :: V
  V = 4d0/3d0*pi*r**3
 end function dvolfunc
 subroutine dvolsub(r,V)
  real(8), intent(in) :: r
  real(8), intent(out) :: V
  V = 4d0/3d0*pi*r**3
 end subroutine dvolsub
end module volmod
```

piについていろいろ検討してみる!

ここでpiを宣言!

main.f90

```
program vol
 real(8) :: r, V
 r = 2d0
 V = dvolfunc(r)
 write(*,*) V
 call dvolsub(r,V)
 write(*,*) V
 write(*,*) pi
 stop
end program vol
```

piは参照できる!!

piは共有変数に!!

例題1

volmod.f90

```
module volmod
 implicit none
 real(8) :: pi = acos(-1d0)
 contains
 function dvolfunc(r) result(V)
  real(8), intent(in) :: r
  real(8) :: V
  V = 4d0/3d0*pi*r**3
 end function dvolfunc
 subroutine dvolsub(r,V)
  real(8), intent(in) :: r
  real(8), intent(out) :: V
  V = 4d0/3d0*pi*r**3
 end subroutine dvolsub
end module volmod
```

ここでpiを宣言!

main.f90

```
program vol
 real(8) :: r, V
 r = 2d0
 V = dvolfunc(r)
 write(*,*) V
 pi = 0d0
 call dvolsub(r,V)
 write(*,*) V
 write(*,*) pi
 stop
end program vol
```

piについていろいろ検討してみる!

piの値を変更

piが変更されたため間違った答えを返す…

piは共有変数に!!

例題1

volmod.f90

```
module volmod
 implicit none
 real(8), parameter :: pi = acos(-1d0)
 contains
 function dvolfunc(r) result(V)
  real(8), int
  real(8) :: V
  V = 4d0/3d
 end function dvolfunc
 subroutine dvolsub(r,V)
  real(8), intent(in) :: r
  real(8), intent(out) :: V
  V = 4d0/3d0*pi*r**3
 end subroutine dvolsub
end module volmod
```

piをparameter属性で宣言!
(変更不可の属性)

piについていろいろ検討してみる!

main.f90

```
implicit none
real(8) :: r, V
r = 2d0
V = dvolfunc(r)
write(*,*) V
pi = 0d0
call dvolsub(r,V)
write(*,*) V
write(*,*) pi
stop
end program vol
```

piはparameterであるため，変更しようとするとError!

Error!!

例題2

rを一次元配列の倍精度実数型の変数とし，適当に値を代入する．そのとき，rを半径とする球の体積を計算する関数とサブルーチンを例題1で書いたモジュールに追加せよ

$$V = \frac{4}{3}\pi r^3$$

例題2

```
module volmod
 implicit none
 real(8), parameter :: pi = acos(-1d0)
 contains
 function dvolfunc(r) result(V)
  real(8), intent(in) :: r
  real(8) :: V
  V = 4d0/3d0*pi*r**3
 end function dvolfunc
 function dmvolfunc(r) result(V)
  real(8), intent(in) :: r(:)
  real(8) :: V(size(r))
  V = 4d0/3d0*pi*r**3
 end function dmvolfunc
(右に続く・・・)
```

```
 subroutine dvolsub(r,V)
  real(8), intent(in) :: r
  real(8), intent(out) :: V
  V = 4d0/3d0*pi*r**3
 end subroutine dvolsub
 subroutine dmvolsub(r,V)
  real(8), intent(in) :: r(:)
  real(8), intent(out) :: V(size(r))
  V = 4d0/3d0*pi*r**3
 end subroutine dmvolsub
end module volmod
```

例題2

関数/サブルーチン名は違うのにする!

配列の宣言は
内部関数のときと同じ!

```
module volmod
 implicit none
 real(8), parameter :: pi = acos(-1d0)
 contains
 function dvolfunc(r) result(V)
  real(8), intent(in) :: r
  real(8) :: V
  V = 4d0/3d0*pi*r**3
 end function dvolfunc
 function dmvolfunc(r) result(V)
  real(8), intent(in) :: r(:)
  real(8) :: V(size(r))
  V = 4d0/3d0*pi*r**3
 end function dmvolfunc
(右に続く・・・)
```

```
 subroutine dvolsub(r,V)
  real(8), intent(in) :: r
  real(8), intent(out) :: V
  V = 4d0/3d0*pi*r**3
 end subroutine dvolsub
 subroutine dmvolsub(r,V)
  real(8), intent(in) :: r(:)
  real(8), intent(out) :: V(size(r))
  V = 4d0/3d0*pi*r**3
 end subroutine dmvolsub
end module volmod
```

配列の宣言は
内部関数のときと同じ!

個別名と総称名

Fortranの関数名には個別名と総称名がある!!

例えば…
sin(x)はxの型がreal(8)でもcomplexでもOK!!
このカラクリは…
・xがreal(8)のときのsin関数は別でdsin
・xがcomplexのときのsin関数は別にcsin
とある. このときdsin, csinは個別名で, 個別名の引数を判断して適切な関数を呼び出すsinを総称名という.

個別名と総称名

Fortranの関数名には個別名と総称名がある!!

例題2で作成したモジュールにはdvolfunc, dmvolfuncがある．これらは個別名であり，別の関数とみなされる．この関数について引数の違い(配列かどうか)を自動で見分けて適切な関数を呼び出す総称名関数volfuncをつくるには，interface文を使う!!

interface文

☆文法

interface 総称名
  module procedure 個別名1, 個別名1, ・・・
end interface 総称名

注意:
 interface文は総称名を定義する以外の機能もあり

例題3

例題2で作成したモジュールの関数dvolfuncとdmvolfuncに総称名volfuncをつけ，サブルーチンdvolsubとdmvolsubに総称名volsubをつけよ.

例題3

```
module volmod
 implicit none
 real(8), parameter :: pi = acos(-1d0)
 interface volfunc
  module procedure dvolfunc,
dmvolfunc
 end interface volfunc
 interface volsub
  module procedure dvolsub,
dmvolsub
 end interface volsub
 contains
 function dvolfunc(r) result(V)
  real(8), intent(in) :: r
  real(8) :: V
  V = 4d0/3d0*pi*r**3
 end function dvolfunc
 (右に続く・・・)
```

ここに追加!!

```
function dmvolfunc(r) result(V)
 real(8), intent(in) :: r(:)
 real(8) :: V(size(r))
 V = 4d0/3d0*pi*r**3
end function dmvolfunc
subroutine dvolsub(r,V)
 real(8), intent(in) :: r
 real(8), intent(out) :: V
 V = 4d0/3d0*pi*r**3
end subroutine dvolsub
subroutine dmvolsub(r,V)
 real(8), intent(in) :: r(:)
 real(8), intent(out) :: V(size(r))
 V = 4d0/3d0*pi*r**3
end subroutine dmvolsub
end module volmod
```