Fortranプログラミング入門

-内部副プログラム(2)-

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.

内部副プログラム

内部副プログラムとは…

- 主プログラムの中に記述する副プログラム
- 内部副プログラムが記述してある主プログラム以外からは利用できない.

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 ← 今回はこれ!!
- 外部副プログラム
- モジュール

関数とサブルーチンの違い

関数：

- 出力結果は基本一つ(関数名やresultによる戻り値)
- 式の中で使える

  例えば y = sin(x) + 10

サブルーチン

- 出力結果はintent属性がout, inoutの変数
- 式の中で使えない
- call文で呼び出す

  例えば call サブルーチン名(引数)

内部サブルーチンの書き方

```
program プログラム名
 宣言部
 実行文
 stop
 contains
 subroutine サブルーチン名 (引数1, 引数2,・・・)
  ・・・
 end subroutineサブルーチン名
end program プログラム名
```

サブルーチンの書き方

☆文法　subroutine サブルーチン名(仮引数1,･･･)

宣言文

実行文

end subroutine サブルーチン名

呼び出し方

call サブルーチン名(引数1,･･･)

例題1

i,j,kを整数型の変数で宣言し，iに適当な値を代入する．そのとき，iに10加えた結果，及び10倍した結果を返すサブルーチンaddtimeを作成せよ．さらに，addtimeを呼び出して，jに10を加えた結果，kに10倍した結果を代入せよ．

例題1

```
program examp1
 implicit none
 integer :: i,j,k
 i = 10
 call addtime(i,j,k)
 write(*,*) "j=",j,"k=",k
 stop
 contains
 subroutine addtime(i,j,k)
 integer, intent(in) :: i
 integer, intent(out) :: j,k
 j = i + 10
 k = i * 10
 end subroutine addtime
end program examp1
```

call文でサブルーチンを呼び出し

contains以下が内部関数・内部サブルーチンの定義

変数iはintent属性がin(入力用の変数)

変数j,kはintent属性がout(出力用の変数)

例題1

```
program examp1
 implicit none
 integer :: i,j,k
 i = 10
 call addtime(i,j,k)
 write(*,*) "j=",j,"k=",k
 stop
 contains
 subroutine addtime(l,m,n)
 integer, intent(in) :: l
 integer, intent(out) :: m,n
 m = l + 10
 n = l * 10
 end subroutine addtime
end program examp1
```

この変数は仮のためi,j,kにする必要はない!