

U-16

**旭川プログラミングコンテスト
資料**

(命令・関数編)

1. 命令・関数とは

HSPにおける命令・関数とは、メインプログラムとは別の場所にてあらかじめ作成した処理を呼び出す機能です。

これはプログラムを作成する時に新しく作ることができます。

プログラムを書く上で同じ処理を何度も書かないように、命令や関数として別の場所を書いてメインプログラムから呼び出すことでプログラムの見た目を良くすることができるほか、プログラムを分けて書くため間違えた場所を直す作業をスムーズに行うことができます。

関数は戻り値と呼ばれる関数の処理を終えた後に呼び出し元に返す値を決める必要があります。命令は戻り値を設定できません。

2. モジュール空間

`#module` `#global`

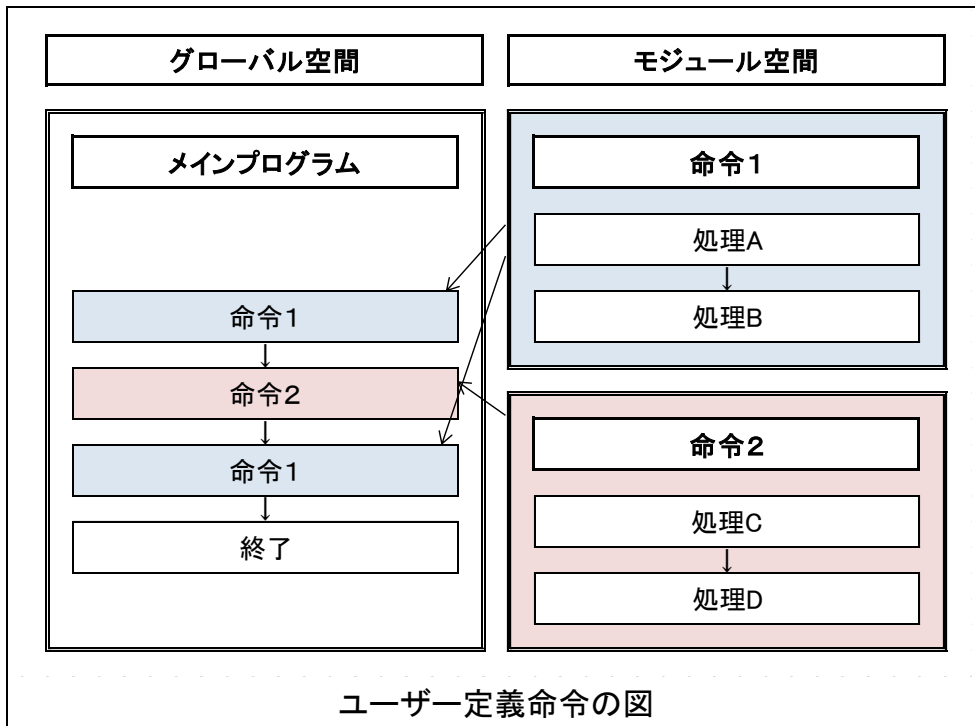
上記 2 つの文字列に囲まれた空間をモジュール空間と呼び、囲まれていない空間をグローバル空間と呼びます。

モジュール空間は複数定義でき、`#module` ~ `#global` 間で囲った数だけ作成されます。

モジュール空間は別の空間にある変数と同じ名前の変数を作っても、異なるものとして扱われる性質があり、

命令や関数を作るときはモジュール空間に書くのが基本です。

グローバル空間に作った場合は基本的にエラーが生じます。



```

#module

#deffunc Hoge int a, int b
    mes "繰り返しはじめ"
    for i,1,b
        mes a * i
    next
    mes "繰り返しおわり"

return
#global

Hoge 10, 3
Hoge 20, 4

```

サンプルプログラム1

```

繰り返しはじめ
10
20
繰り返しおわり
繰り返しはじめ
20
40
60
繰り返しおわり

```

サンプルプログラム1の実行結果

3. 命令・関数の作成

#deffunc ~ return

新しく命令を作ることができます。命令の中には、他のモジュール空間で作成された命令や関数も使用することができます。

呼び出すときは、サンプルプログラム1の「Hoge 10,3」のように、

命令の名前 パラメータ

の形で呼び出し、複数パラメータがある時はカンマ(,)で区切ります。

#deffunc の後に型(変数の種類)とパラメータを指定することによって、作成した命令に対して命令の中で外部の変数の値を用いることができます。これを引数といいます。

型の名前とパラメータの種類は以下の表の通りです。

型の名前	パラメータの種類
int	整数値
double	実数値
array	配列
str	文字列

#defcfunc ~ return

新しく関数を作ることができます。関数は、returnの後に返したい値を設定する必要があります。呼び出すときは、サンプルプログラム2の「Hoga (おはよう,ございます)」のように、

関数の名前 (パラメータ)

の形で呼び出し、複数パラメータがある時はカンマ(,)で区切ります。

#deffunc 同様に、#defcfunc の後に型とパラメータを指定することによって関数の中で外部の変数の値を用いることができます。

4. 注意点

- ・ モジュール空間の中で新しく用意した変数は、モジュール空間の外で同様に扱うことはできません。
- ・ #defcfunc では必ず return の後に戻り値を設定しなければいけません。一方、#deffunc は戻り値を設定できません。
- ・ 関数、命令で定義した引数に対し、値や文字を代入することはできません。

```
#module

#defcfunc Hoga str a, str b
    mes "「" + a + "」と「" + b + "」を合わせます"
return (a + b)                ;(a + b)が戻り値
```

#global

```
c = Hoga("おはよう","ございます")
```

```
mes c
```

サンプルプログラム2

```
「おはよう」と「ございます」を合わせます
```

```
おはようございます
```

サンプルプログラム2の実行結果

5. 練習問題

1. 2つの整数を渡すと足した数、引いた数、かけた数を表示してくれる命令を作ってみよう
2. 2つの文字を渡すと1つ目の文字と2つ目の文字を交互に4回表示してくれる命令を作ってみよう。
3. 1つの文字を渡すと鍵括弧(「」)で閉じて戻り値として返してくれる関数をつくってみよう
4. 4つの整数を渡すと、戻り値に
 - ① すべて足して20より大きいなら「当たり」
 - ② すべて足して20なら「大当たり」
 - ③ すべて足して20未満なら「はずれ」を返してくれる関数を作ってみよう