

関数とサブルーチン

一連の処理を一つの部品とする

Pascal

関数(function)

手続き(procedure)

FORTTRAN

関数副プログラム

サブルーチン副プログラム

c言語

関数

関数

数学

$$z = f(x, y)$$

f: 数 x, y に対してある数に対応させる規則
対応した数(f(x,y))と数 z が等しい

プログラミング言語

$$z := f(x, y)$$

データ x, y を用いてある処理をし、その結果に依存する「値」を変数 z に代入する

例題

実数を二つ読み込んでその最大値を出力する

```
program showmax(input, output);
var x, y, z: real;
begin
  read(x, y);
  if x > y then z := x else z := y;
  write (z)
end.
```

z := mmax(x, y);

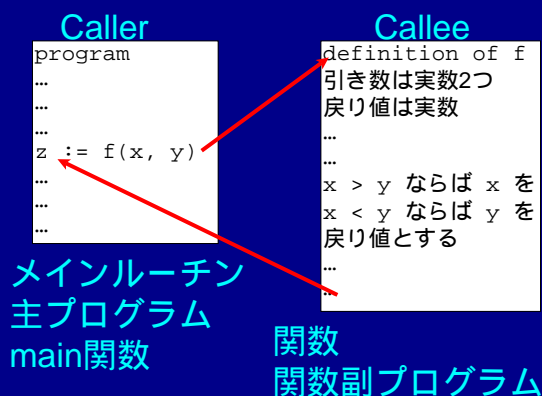
関数

$$z := f(x, y)$$

x, y: 引き数(arguments)

zに代入される「値」: 戻り値(return value)

関数



例題(Pascal)

```
program showmax(input, output);
var a, b, z: real;
function mmax(x, y: real): real;
begin
  if x > y then mmax := x
  else mmax := y;
end;
begin
  read(a, b);
  z := mmax(a, b);
  write (z)
end.
```

関数定義(Pascal)

```
program プログラム名(input, output);
const 定数宣言;
var 変数宣言;
function 関数名(引数宣言): 関数型;
  const 局所定数宣言;
  var 局所変数宣言;
begin
  ...
  関数名 := 戻り値;
end;
begin ... end.
```

例題(FORTTRAN)

PROGRAM SHWMAX	REAL FUNCTION MMAX(X, Y)
REAL A, B, Z, MMAX	REAL X, Y
READ(5, *) A, B	IF (X.GT.Y) THEN
Z = MMAX(A, B)	MMAX = X
WRITE(6, *) Z	ELSE
STOP	MMAX = Y
END	END IF
	RETURN
	END

関数定義(FORTRAN)

Caller側

関数型 関数名

関数型 FUNCTION 関数名(引数名)
局所定数宣言 and/or 局所変数宣言
引数型 引数名

...

関数名 = 戻り値

RETURN

END

例題(c言語)

```
#include <stdio.h>
main () {
    double a, b, z, mmax();
    scanf("%lf %lf", &a, &b);
    z = mmaz(a, b);
    printf("%lf\n", z);
}
double mmax(x, y)
double x, y;
{
    if (x > y) return x;
    else return y
}
```

例題(c言語)

```
#include <stdio.h>
double mmax(double s, double t);
int main (void) {
    double a, b, z;
    scanf("%lf %lf", &a, &b);
    z = mmaz(a, b);
    printf("%lf\n", z);
    return 0;
}
double mmax(double x, double y) {
    if (x > y) return x;
    else return y
}
```

関数定義(c言語)

main関数の前 or Caller内

関数型 関数名(引数型 引数名);
/* 関数プロトタイプ宣言 */

```
関数型 関数名(引数型 引数名){
    局所定数宣言 and/or 局所変数宣言
    ...
    return 戻り値;
}
```

標準関数(Pascal)

abs(x)	x	絶対値
sqr(x)	x^2	自乗
sqrt(x)	\sqrt{x}	平方根
sin(x)	$\sin x$	正弦
cos(x)	$\cos x$	余弦
arctan(x)	$\tan^{-1}x$	逆正接
exp(x)	e^x	指数関数
ln(x)	$\log_e x$	自然対数

組み込み関数(FORTRAN)

INT(x)	実数 整数	SINH(x)	双曲線関数
FLOAT(x)	整数 実数	COSH(x)	
ABS(x)	絶対値	TANH(x)	
MOD(x,y)	x/yの剰余	EXP(x)	指数関数
SQRT(x)	平方根	LOG(x)	自然対数
SIN(x)	三角関数	LOG10(x)	10を底の対数
COS(x)			
TAN(x)			
ASIN(x)	逆三角関数		
ACOS(x)			
ATAN(x)			
ATAN2(x,y)	ATAN(x/y)		

標準ライブラリ関数(c言語)

math.h (STRICT ANSI)

cos(x)	三角関数	sqrt(x)	平方根
sin(x)		fabs(x)	絶対値
tan(x)		exp(x)	指数
acos(x)	逆三角関数	pow(x,y)	x^y
asin(x)		log(x)	対数
atan(x)		log10(x)	
atan2(y,x)			
cosh(x)	双曲線関数		
sinh(x)			
tanh(x)			

例題(関数と配列)

実数値を10個読み込んでその平均値を出力する

```
const n=10
var x: array[1..n] of real;
    s: real;

s := 0.0;
for i:=1 to n do s := s + x[i];
s := s/n;
```

例題(Pascal)

```
program showave(input,output);
const n = 10;
var a: real; x: array [1..n] of
real; i: integer;
function getave(y: array [1..n] of
real): real;
var s: real; j: integer;
begin
s:=0;
for j:=1 to n do s:=s + y[j];
getave := s/n;
end;
```

例題(Pascal)

```
program showave(input,output);
const n = 10;
type index = [1..n];
tdata = array [index] of real;
var a:real; x:tdata; i:index;
function getave(y: tdata): real;
var s: real; j: index;
begin
s:=0;
for j:=1 to n do s:=s + y[j];
getave := s/n;
end;
```

例題(Pascal)

```
begin
for i:= 1 to n do read(x[i]);
a := getave(y);
writeln(a)
end.
```

関数と配列(Pascal)

変数 := 関数名(配列名, ...);

(例)

```
type mdata:array[1..n] of real;
var amatrix:mdata;
avalue:real
function afun(amtrix:mdata):real;...
function bfun(aval:real):real;...

avalue := afun(amatrix);
avalue := bfun(amatrix[3]);
```

例題(FORTRAN)

```
PROGRAM SHOWAV
PARAMETER(N=10)
INTEGER I
REAL A, X, GETAVE
DIMENSION X(1:N)
DO 10 I = 1, N
READ(5,*) X(I)
10 CONTINUE
A = GETAVE(X, N)
WRITE(6,*) A
STOP
END
```

例題(FORTRAN)

```
REAL FUNCTION GETAVE(Y, K)
INTEGER I
REAL S, Y
DIMENSION Y(1:K)
S = 0.0
DO 10 I = 1, K
S = S + Y(I)
10 CONTINUE
GETAVE = S/FLOAT(K)
RETURN
END
```

関数と配列(FORTRAN)

整合配列と整合寸法

FUNCTION FNAME(MNAME, NUMBER)

DIMENSION MNAME(NUMBER)

例題(c言語)

```
#include <stdio.h>
#define N 10
double getave(double *z, int n);
int main(void){
double a, x[N];
int i;
for(i=0;i<N;i++){
scanf("%lf",&x[i]);
}
a = getave(x, N);
printf("%lf\n",a);
return 0;
}
```

例題(c言語)

```
double getave(double *y, int k){
    int i;
    double s;
    s = 0.0;
    for(i=0;i<k;i++){
        s+=y[i];
    }
    s /= (double) k;
    return s;
}

for(i=0;i<k;i++) s+=*y++;
```

関数と配列(c言語)

```
double getave(double y[]);
double getave(double y[N]);
double getave(double *y);
double getave(const double *y);
```

y[0], y[1], ...

*y