

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)

FORTRAN

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

サブルーチン

関数のように値を返さず処理のみ行う

アドレス参照を用いることにより複数のデータを呼び出し元に返すこともできる

例題

10個の実数を読み込み最大値、最小値を出力する

```
program maxmin(input,output);
const n=10;
var x:array [1..n]:real;
    max, min:real;
    i: integer;
begin
  for i:=1 to n do read(x[i]);
  max := x[1];
  min := x[1];
  for i:=2 to n do
    begin
      if x[i] > max then max:=x[i];
      if x[i] < min then min:=x[i]
    end;
  writeln(min, max)
end.
```

関数(1)

```
program maxmin(input,output);
const n=10;
var x:array [1..n]:real;
    max,min:real; i: integer;
function getmax(y:array[1..n]:real):real;
var i:integer; mx: real;
begin
  mx:=y[1];
  for i:=2 to n do
    if y[i]>mx then mx:=y[i];
  getmax:=mx
end;
```

関数(2)

```
function getmin(z:array[1..n]:real):real;
var i:integer; mn: real;
begin
  mn:=z[1];
  for i:=2 to n do
    if z[i]<mn then mn:=z[i];
  getmin:=mn
end;
begin
  for i:=1 to n do read(x[i]);
  max := getmax(x);
  min := getmin(x);
  writeln(min, max)
end.
```

```
program showmaxmin(input,output);
const n=10;
type vect:array [1..n]:real;
var x: vect; max,min:real;i:integer;
procedure minmax(y:vect; var mx: real; var
min:real);
var i:integer;
begin
  mx:=y[1];
  mn:=y[1];
  for i:=2 to n do
    begin
      if y[i]>mx then mx:=y[i];
      if y[i]<mn then mn:=y[i]
    end
  end;
end;
```

```
begin
  for i:=1 to n do read(x[i]);
  minmax(x, min, max);
  writeln(min, max)
end.
```

手続き(Pascal)

procedure 手続き名(引数宣言);

```
program p_name(input,output);
const c_name=constant_value;
var v_name: variable_type;
function f_name(arg_name:arg_type):function_type;
const c_name=constant_value;
var v_name: variable_type;
begin ... f_name:=return_value end;
procedure f_name(arg_name:arg_type);
const c_name=constant_value;
var v_name: variable_type;
begin ... end;
```

値仮引数と参照仮引数

procedure p_name(val:type);
valは手続き中でその値を変更してもCaller側の実引数には影響しない
CALL BY VALUE (値参照)

procedure p_name(var val:type);
valは手続き中でその値を変更した場合Caller側の実引数も同時に変更される
CALL BY REFERENCE (アドレス参照)

例題(FORTRAN)

```
PROGRAM SHMXMN
PARAMETER(N=10)
REAL X,MX,MN
INTEGER I
DIMENSION X(1:N)
READ(5,*) (X(I),I=1,N)
CALL MINMAX(X,N,MX,MY)
WRITE(6,*) MX,MN
STOP
END
```

例題(FORTRAN)

```
SUBROUTINE MINMAX(Y,M,MX,MY)
REAL X,MX,MY
INTEGER I
DIMENSION Y(1:M)
MX=Y(1)
MY=Y(1)
DO 10 I=2,M
  IF(MX.GT.Y(I)) MX=Y(I)
  IF(MY.LT.Y(I)) MY=Y(I)
10 CONTINUE
RETURN
END
```

サブルーチン副プログラム

(FORTRAN)

SUBROUTINE 名前(仮引数)

```
PROGRAM p_name
STOP
END
FUNCTION f_name(args)
f_name:=return_value
return
end
SUBROUTINE s_name(args)
return
end
```

サブルーチン副プログラム

(FORTRAN) CALLER側

CALL サブルーチン名(実引数)

値仮引数と参照仮引数

SUBROUTINE p_name(val)

valはサブルーチン内でその値を変更した場合Caller側の実引数も同時に変更される

FORTRANは常にCALL BY REFERENCE (アドレス参照)でCall by valueの利用法はできない

例題(c言語)

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

```

void minmax(double *x, int m, double *a,
double *b){
    int i;
    *a=x[0];
    *b=x[0];
    for (i=1;i<m;i++){
        if(*a>x[i])*a=x[i];
        if(*b<x[i])*b=x[i];
    }
}

```

void型関数

```

void 関数名(引数型 仮引数);
    プロトタイプ宣言
void 関数名(引数型 仮引数){...}

```

```

#include ...
#define ...
void f_name(args_type args);
int main(void){
    return ...;
}
void f_name(args_type args){
    ...
}

```

値仮引数と参照仮引数

type p_name(type val)

valは手続き中でその値を変更してもCaller側の実引数は変更されない

c言語は常にCALL BY VALUE (値参照)

Call by referenceの利用法をする場合ポインタ変数を使用する

Call By Value

```

program ex(output);
var a:real;
procedure p(x:real);
begin
    x:=x+1.
end;
begin
    a:=1.;
    p(a);
    writeln(a)
end.

```

1.0が出力される

Call By Reference

```

program ex(output);
var a:real;
procedure p(var x:real);
begin
    x:=x+1.
end;
begin
    a:=1.;
    p(a);
    writeln(a)
end.

```

2.0が出力される

Call By Value

```

#include <stdio.h>
void p(double x);
int main(void){
    double a;
    a=1.;
    p(a);
    printf("%lf\n",a);
    return 0;
}
void p(double x){
    x += 1.;
}

```

1.0が出力される

Call By Reference

```

#include <stdio.h>
void p(double *x);
int main(void){
    double a;
    a=1.;
    p(&a);
    printf("%lf\n",a);
    return 0;
}
void p(double *x){
    *x += 1.;
}

```

2.0が出力される

Call By Reference

```

PROGRAM EX
REAL*8 A
A=1.D
CALL P(A)
WRITE (6, *) A
STOP
END
SUBROUTINE P(X)
REAL*8 X
X=X+1.D0
RETURN
END

```

2.0が出力される

まとめ

FORTRAN

常にCall by reference

引き数の管理はユーザが行う

まとめ

Pascal

仮引数:型; Call by value

var 仮引数:型; Call by reference

引き数の管理はコンパイラが行う

まとめ

C言語

常にCall by value

ポインタを使うことによって疑似的にCall by referenceを行うことができる

配列の引き数は常にポインタ

引き数の管理はプロトタイプ宣言によりコンパイラが行うこともできる

大域変数と局所変数

```
program ...
var v: ...
    i, k, n: ...
function f(v:...;n:...):...;
    var i, m: ...;
    begin
        ...
    end;
begin
    ...
end.
```

大域変数

局所変数

大域変数と局所変数

```
#include ...
integer k;
int main(void){
    double v[100];
    int i, n;
    f(v, n);
...
void
f(double v[],int n){
    int i, m;
    ...
}
```

大域変数

局所変数

大域変数と局所変数

```
PROGRAM ...
COMMON /AREA/X,Y,Z
SUBROUTINE ...
COMMON /AREA/A,B,C
```

共通ブロック名

例題

配列に格納された整数のうち最大値を与えるものの添字番号と最大値を表示する

x[0]	x[1]	x[2]	x[3]	x[4]	x[5]	x[6]
60	23	56	76	32	44	68

3 76

例題(単純な関数)

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

単純な関数

```
int argmax(int x[], int n){
    int i, k, max;
    k = 0;
    max = x[k];
    for (i = 1; i < N; i++){
        if (x[i] > max) {
            k = i;
            max = x[i];
        }
    }
    return k;
}
```

汎用的な関数

与えられた添え字の範囲内で最大の要素の添え字を返す

```
int argmax(int x[], int ifirst, int iend){
    int i, k, max;
    k = ifirst;
    max = x[k];
    for (i = ifirst + 1; i <= iend; i++){
        if (x[i] > max) {
            k = i;
            max = x[i];
        }
    }
    return k;
}
```

例題(汎用的な関数)

```
#include <stdio.h>
#define N 10
int argmax(int x[], int ifirst, int iend);
int main(void){
    int i, imax, x[N];
    for(i=0;i<N;i++)scanf("%d",&x[i]);
    imax = argmax(x, 0, N-1);
    printf("%d %d\n", imax, x[imax]);
    return 0;
}
```

例題(並べ直して出力)

```
#include <stdio.h>
#define N 10
int argmax(int x[], int ifirst, int iend);
void swap(int *a, int *b);
int main(void){
    int i, imax, x[N];
    for(i=0;i<N;i++)scanf("%d",&x[i]);
    for(i=0;i<N;i++){
        imax = argmax(x, i, N-1);
        swap(&x[imax], &x[i]);
    }
    for(i=0;i<N;i++) printf("%d\n", x[i]);
    return 0;
}
```

データの入れ替え

```
void swap(int *a, int *b)
{
    int tmp;
    tmp = *a;
    *a = *b;
    *b = tmp;
}
```