

制御構造

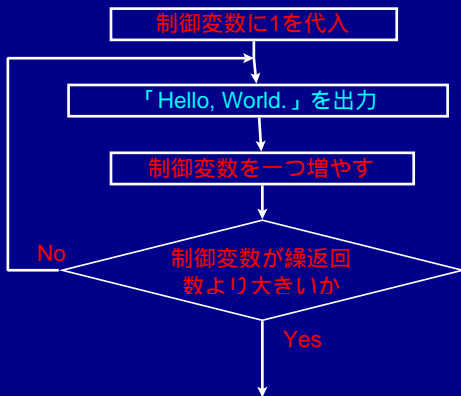
決まった回数の繰り返し
条件判断
回数の決まっていない繰り返し

【例題1】

「Hello, World」を三回表示するプログラム

```
program hello3(output);  
begin  
  writeln('Hello, World.');
```

決まった回数の繰り返し



Pascal

```
program hello3(output);  
const n = 3;  
var i: interger;  
begin  
  for i:=1 to n do  
    writeln('Hello, World.')
```

FORTRAN

```
PROGRAM HELLO3  
PARAMETER (N = 3)  
INTEGER I  
DO 10 I=1, N  
  WRITE(6, *)'Hello, World.'  
10 CONTINUE  
STOP  
END
```

C

```
#include <stdio.h>  
#define N 3  
int main(void) {  
  int i;  
  for (i=0; i<N; i++) {  
    printf("Hello, world.¥n");  
  }  
  return 0;  
}
```

【例題1】

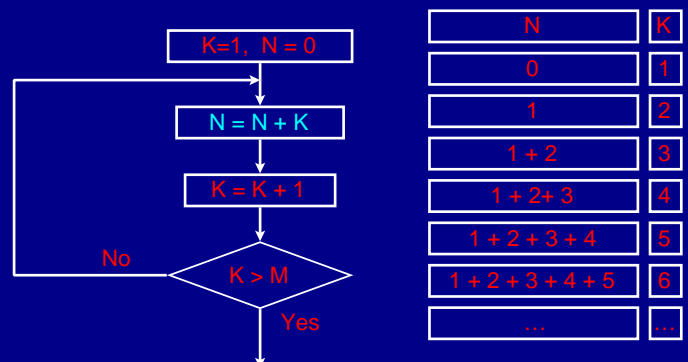
正の整数 M を読み込み
を計算し出力するプログラム

$$N = \sum_{k=1}^M k = \frac{1}{2} M(M+1)$$

$$N = \prod_{k=1}^M k = M!$$

計算上のテクニック

$$N = 1 + 2 + 3 + \dots + M$$



Pascal

```
program addm(input, output);
var i, m, n: interger;
begin
  read(n);
  m := 0;
  for i := 1 to n do
    begin
      m:=m + i;
      writeln(i, m)
    end
  end.
end.
```

FORTRAN

```
PROGRAM ADDM
INTEGER I, M, N
READ(5,*) N
M = 0
DO 10 I = 1, N
M = M + I
WRITE(6,*) I, M
10 CONTINUE
STOP
END
```

C

```
#include <stdio.h>

int main(void){
  int i, m, n;
  scanf("%d", &n);
  m = 0;
  for (i = 1; i <= n; i++) {
    m += i;
    printf("%d %d\n", i, m);
  }
  return 0;
}
```

演算子の短縮記法(c言語)

k = k + 1	k++
k = k - 1	k--
k = k + m	k+=m
k = k - n	k-=n
k = k * j	k*=j
k = k / i	k/=i

条件判断(c言語)

回数の決まった繰り返しに用いるもののみ

a ≤ b	a<=b
a < b	a<b
a ≥ b	a>=b
a > b	a>b

Pascal

```
program multim(input, output);
var i, m, n: interger;
begin
  read(n);
  m := 1;
  for i := 1 to n do
    begin
      m:=m*i;
      writeln(i, m)
    end
  end.
end.
```

FORTRAN

```
PROGRAM MULTIM
INTEGER I, M, N
READ(5,*) N
M = 1
DO 10 I = 1, N
M = M*I
WRITE(6,*) I, M
10 CONTINUE
STOP
END
```

C

```
#include <stdio.h>

int main(void){
  int i, m, n;
  scanf("%d", &n);
  m = 1;
  for (i = 1; i <= n; i++) {
    m *= i;
    printf("%d %d\n", i, m);
  }
  return 0;
}
```

回数の決まった繰り返し(Pascal)

for 変数 := 式 to 式 do ブロック

制御変数 初期値 終了値

cf) for 変数 := 式 downto 式 do

回数の決まった繰り返し(Pascal)

for 変数 := 式 to(downto) 式 do

例

```
for i:=1 to 3*n do begin ... end;
  i=1,2,3,4 ... 3n-2,3n-1,3n
for j:=-3 to 10 do begin ... end;
  j =-3,-2,-1,0,1,2,3 ... 8,9,10
for k:=4*n downto 2*n do begin ...
  k = 4n, 4n-1, 4n-2, ..., 2n+2,
  2n+1, 2n
```

定回数の繰り返し(FORTRAN)

DO 文番号 変数 = 式, 式 (, 式)

制御変数 初期値 終了値 増分

制御変数は整数型(実数型)

初期値、終了値、増分は制御変数と同じ型

文番号はcontinue文に付けるのが望ましい

途中で制御変数を変更してはならない

定回数の繰り返し(FORTRAN)

例

```
DO 10 I= 1,3*n
  I = 1, 2, 3, 4, 5, ..., 3n-2, 3n-1 3n
DO 10 j = -3, 10, 2
  j = -3, -1, 1, 3, 5, 7, 9
DO 10 k = 4*n, 2*n, -1
  k = 4n, 4n-1, 4n-2, ..., 2n+2, 2n+1, 2n
DO 10 x = 0.0, 0.1, 0.01
  x=0.00, 0.01, 0.02, 0.03, ..., 0.09, 0.10
```

回数の決まった繰り返し(c)

言語仕様の中に存在しない

(forで代用)

do 文番号 変数=初期値, 終了値, 増分

for(変数=初期値; 変数<=終了値;変数+=増分)

回数の決まった繰り返し(c)

言語仕様の中に存在しない

(forで代用)

do 文番号 変数=初期値, 終了値, 増分

for(変数=初期値; 変数<=終了値;変数+=増分)

制御変数は整数型だけを使う

繰り返すブロックの中で制御変数を変更し

ない

回数の決まった繰り返し(c)

例

```
for (i = 0; i<3*n; i++){
  i = 0, 1, 2, 3, 4, 5, ..., 3n-2, 3n-1
for (j = -3; j <= 10; j+=2){
  j = -3, -1, 1, 3, 5, 7, 9
for (k = 4*n; k >= 2*n; k--){
  k = 4n, 4n-1, 4n-2, ..., 2n+2, 2n+1, 2n
```