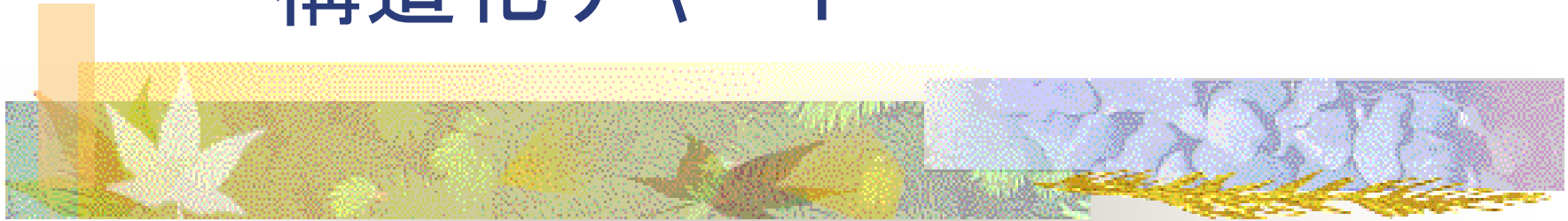


アルゴリズムとデータ構造a

3 - 構造化プログラミング、 構造化チャート



大見 嘉弘



今日の授業

- 構造化プログラミング
 - 概念と特徴
 - 順次構造
 - 選択構造
 - 繰り返し構造
- フローチャートと構造化
- 構造化チャート



構造化プログラミング

- プログラムの構造にルールを決めて、明確で、正確、読みやすい、保守しやすいプログラムを作る。
- 生産性と品質を高めることができる
- 共同作業の効率や保守性も向上

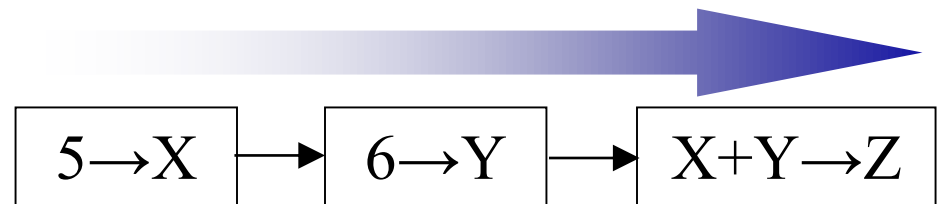
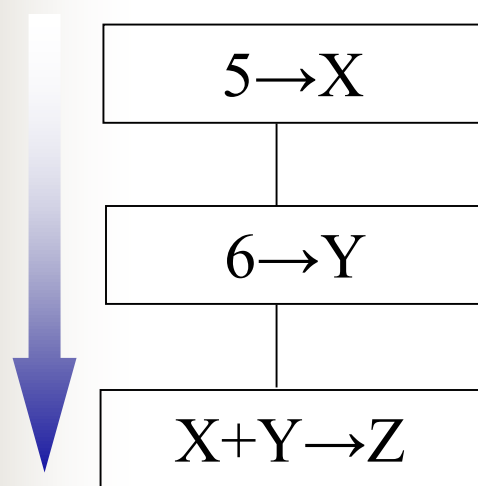


構造化プログラミングのルール

- **順次、選択、繰返し**の3つの構造だけでプログラムを作る
- 構造化定理
1つの入口と1つの出口を持つプログラムなら、順次、選択、繰返しの構造の組合せで記述できる

順次構造(Sequence)

- 1つの処理が終了したら、次に処理に進む
- 単に順番に処理が進む



左→右は矢印をつけたほうが良い

上→下は矢印をつけないことが多い

処理の書き方

Xに1を代入

$1 \rightarrow X$

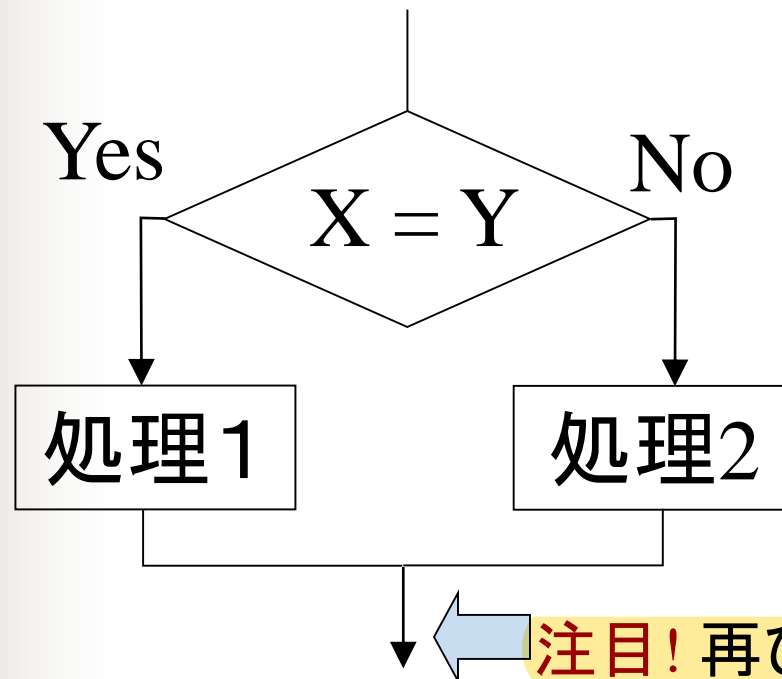
$X \leftarrow 1$

同じ処理

- 流れ図の代入は “ \leftarrow ”
- Javaの場合 $x = 1;$
- まれに、流れ図で代入を $=$ で書いているものもある。紛らわしいので望ましくない

選択構造(Selection)

- ある条件によって、2つの処理のうちどちらかを選択して実行する



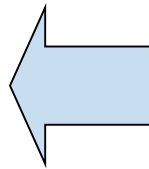
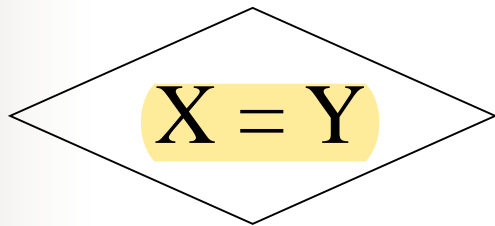
◆ $X=Y$ の時、処理1を実行

◆ $X \neq Y$ の時、処理2を実行

Javaならif文やswitch文

注目! 再び1つにまとまる

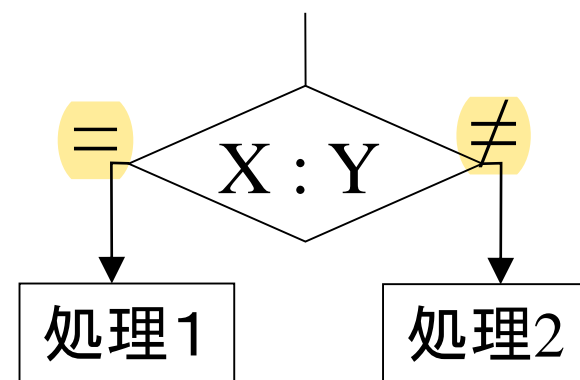
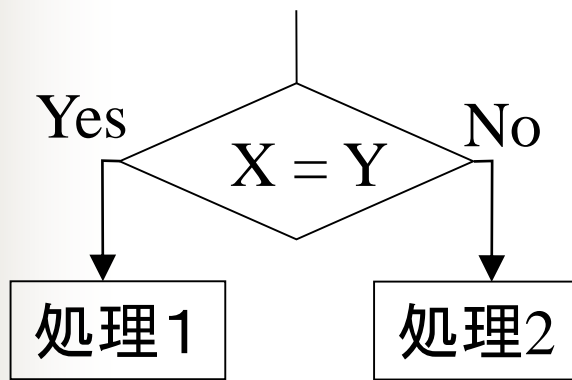
条件の書き方



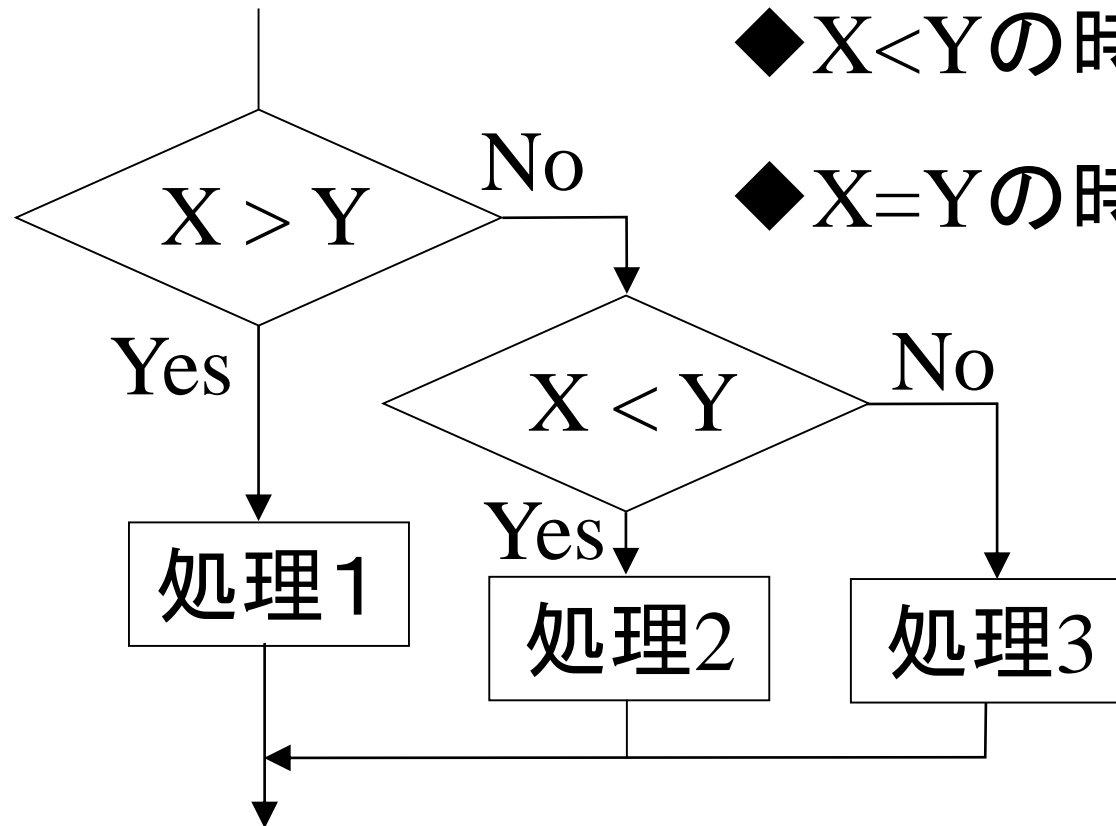
XとYが等しいかどうか

$X = Y$ とは書かない!

■ Javaの場合、`if (x == y) { ... }`



多段分岐

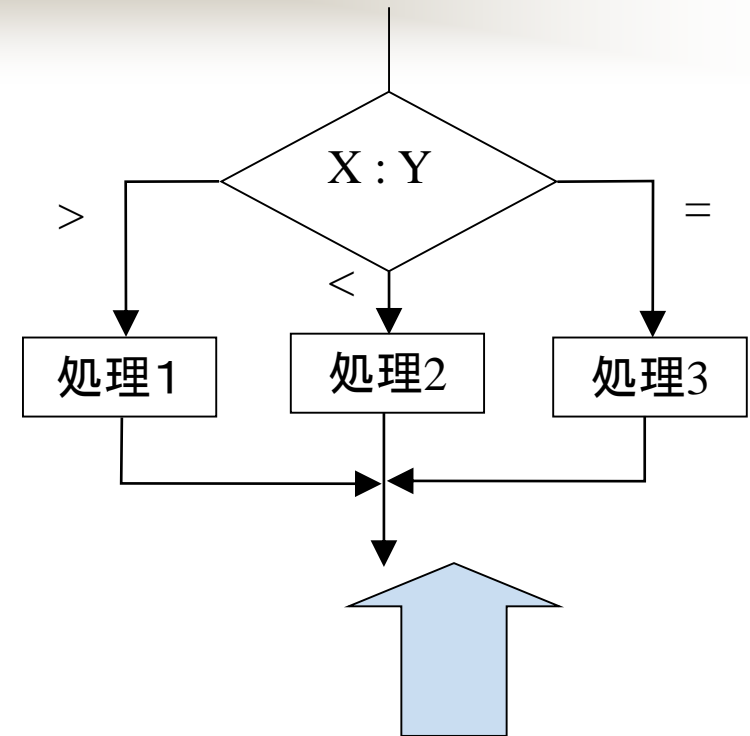
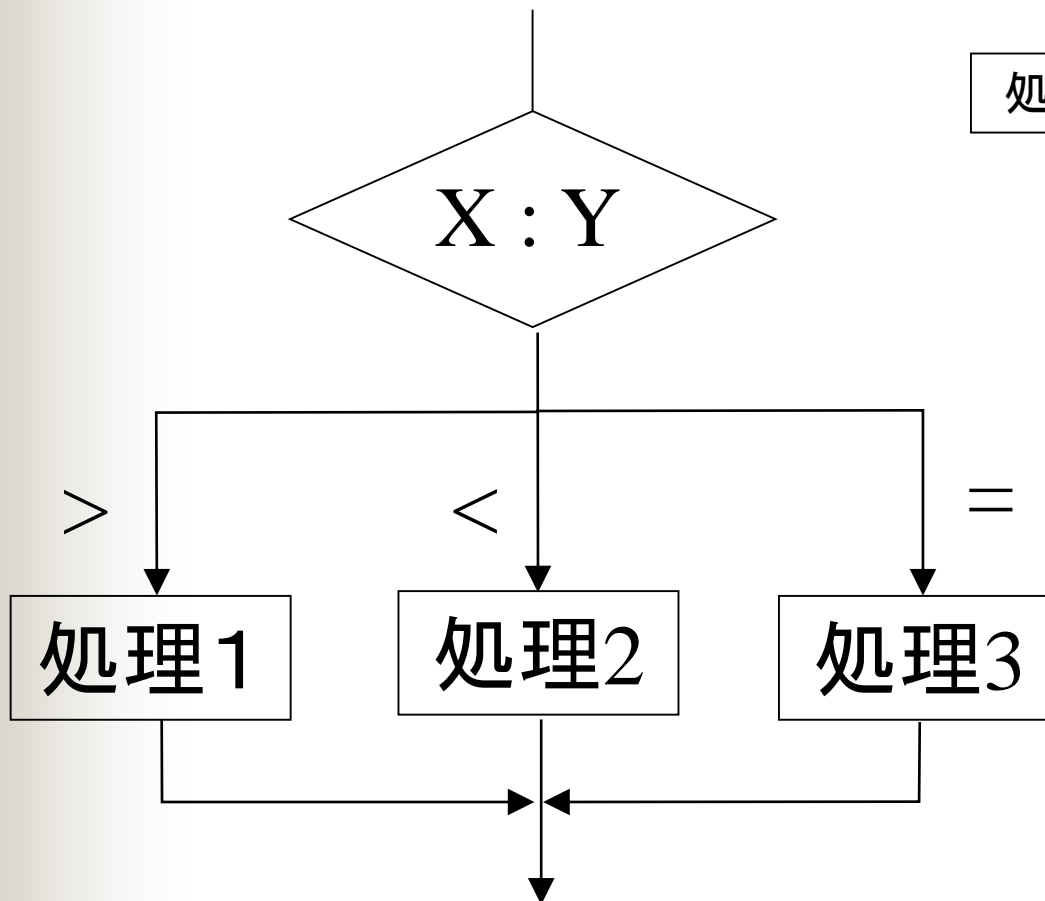


◆ $X > Y$ の時、処理1

◆ $X < Y$ の時、処理2

◆ $X = Y$ の時、処理3

多分岐記号



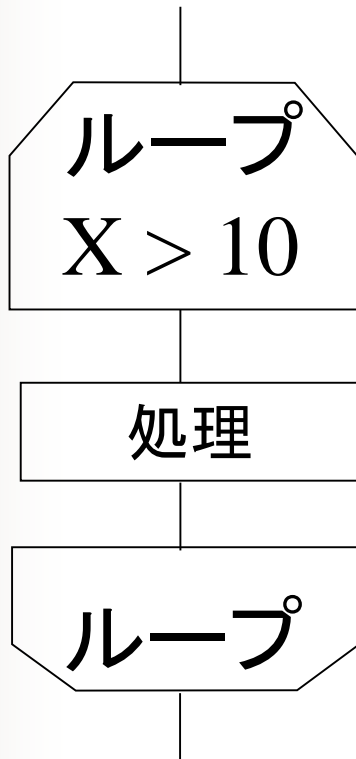
3分岐の場合、
これでも可



繰返し構造(Iteration)

- ある条件を満たす限り、処理を繰返し実行
- 条件を満たさなくなったら、次の処理に進む
- Javaのfor文やwhile文、do while文に相当
- ただし... JISの流れ図を書く場合は、
 - ある条件を満たさない限り、繰り返す
 - 条件を満たしたら、次の処理に進む
 - 条件の関係が反対！

JIS流れ図の繰返し (前条件)

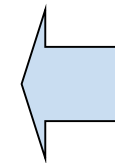


◆ $X > 10$ でない時 ($X \leq 10$)、
処理を繰返す

◆ $X > 10$ になったら、繰返し
をやめて、次に進む

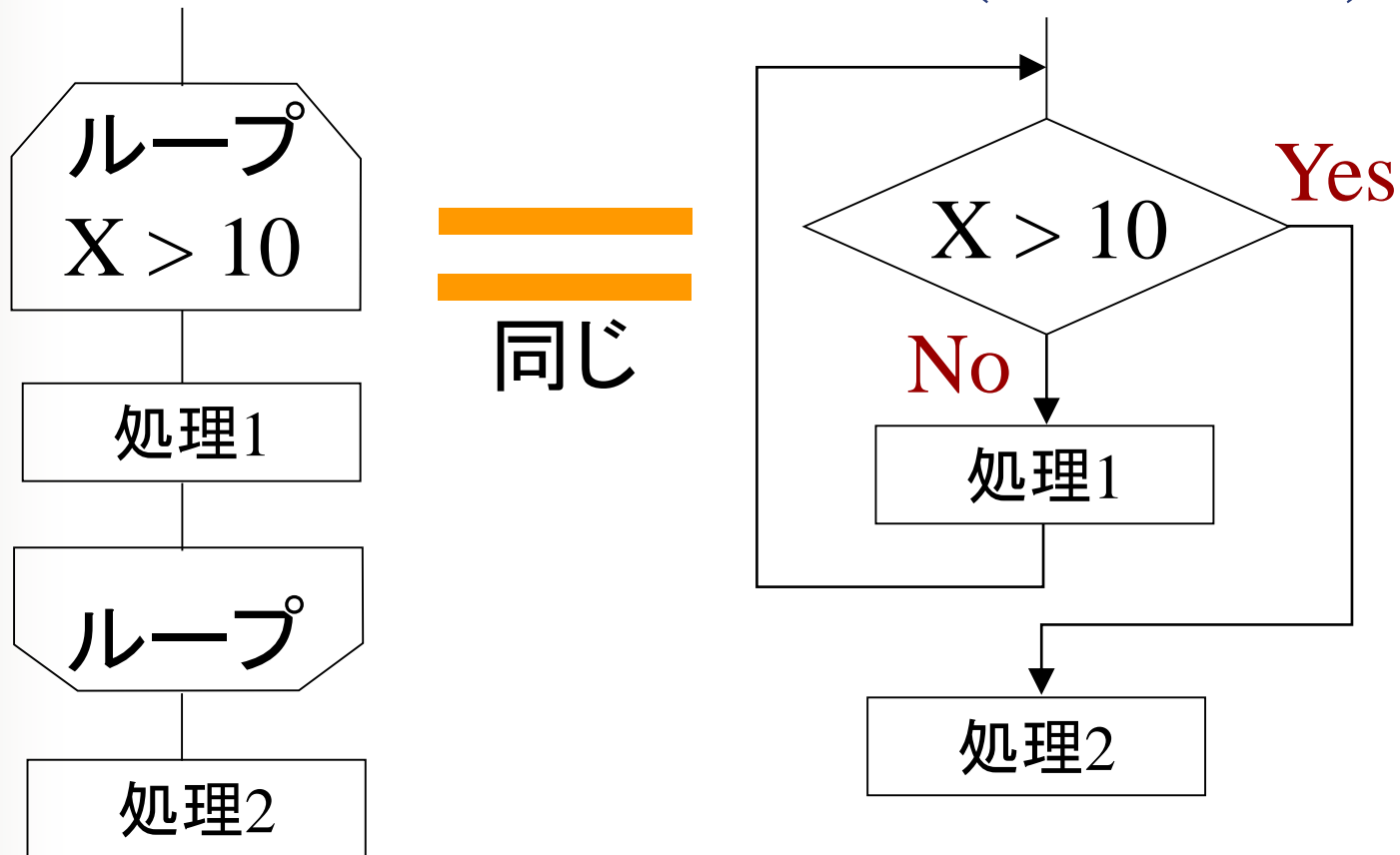
Javaの場合、

```
while (x <= 10) {  
    処理;  
}
```

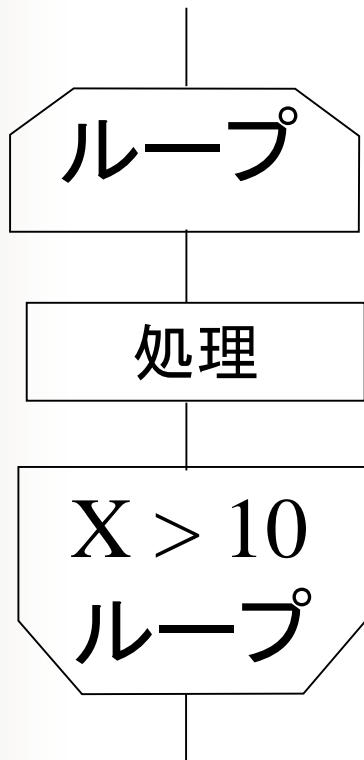


条件が
反対！

JIS流れ図の繰返し (前条件) (2)



JIS流れ図の繰返し (後条件)



◆必ず1度は処理を行う

◆ $X > 10$ でないなら($X \leq 10$)、処理を繰返す

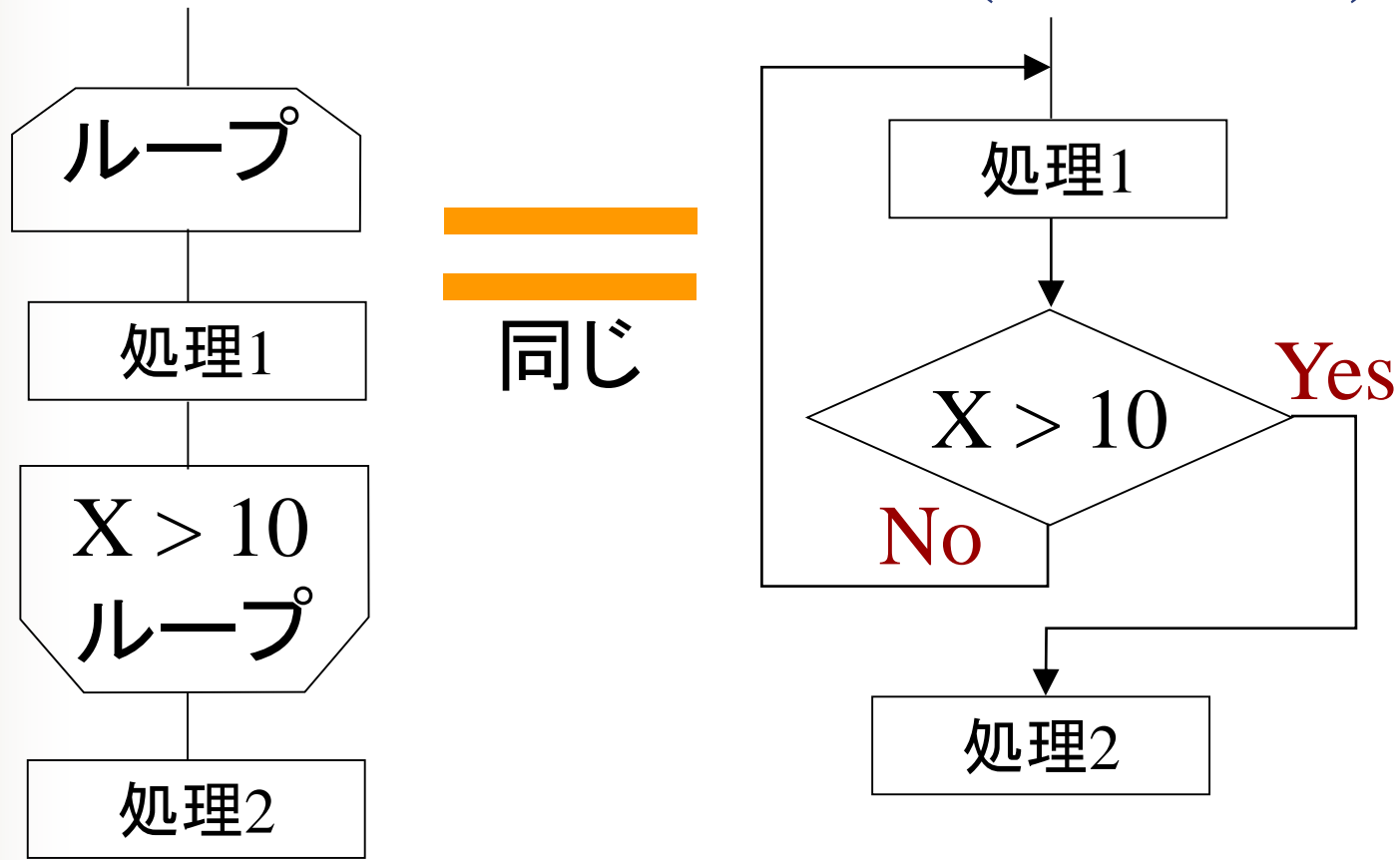
◆ $X > 10$ になったら、次に進む

Javaの場合、

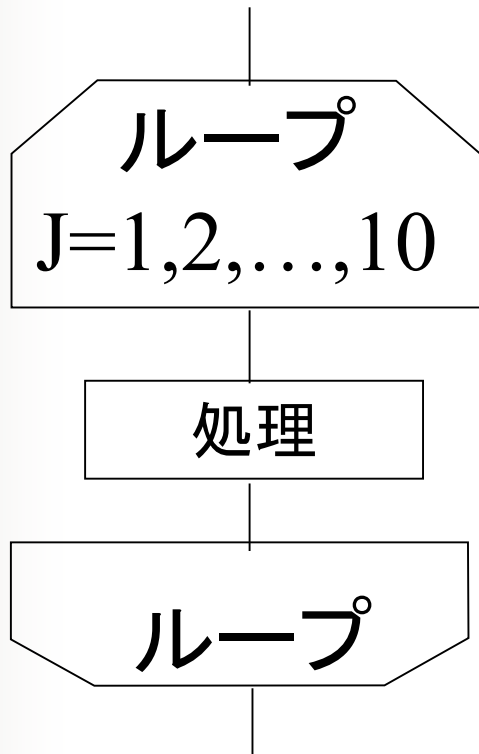
```
do {  
    処理;  
} while (x <= 10);
```

条件が
反対!

JIS流れ図の繰返し(後条件)(2)



繰返し回数の指定

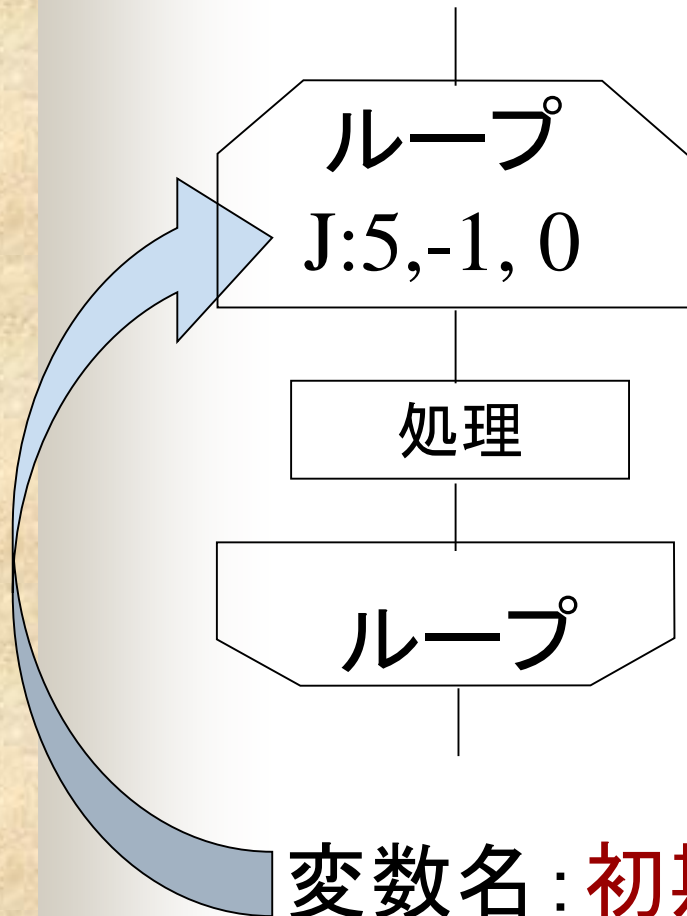


- J=1, J=2, J=3, ..., J=10と10まで繰返す
- つまり10回繰返す

Javaの場合、

```
for (j = 1; j <= 10; j++) {  
    処理;  
}
```

初期値、増分、終了値の指定



- J=5から始まり、
J=4,J=3,..., J=0と繰返す
- つまり6回繰返す

Javaの場合、

```
for (j = 5; j >=0; j--) {  
    処理;  
}
```

変数名 : 初期値、増分、終了値

繰返しの入れ子

ループ1
 $i=1,2,\dots,10$

ループ2
 $j=1,2,\dots,15$

処理

ループ2

ループ1

- ループ1の中にループ2が入っている
- 多重ループ

Javaの場合、

```
for (i = 1; i <= 10; i++) {  
    for (j = 1; j <= 15; j++) {  
        処理;  
    }  
}
```

プログラミングの歴史

■ 構造化プログラミング以前

- goto文 : プログラムの自由な位置に飛べる
- 機械語のjump命令がルーツ
- 好きなようにプログラムが書けるが、処理の流れがぐちゃぐちゃになりやすい
→ スパゲティプログラム

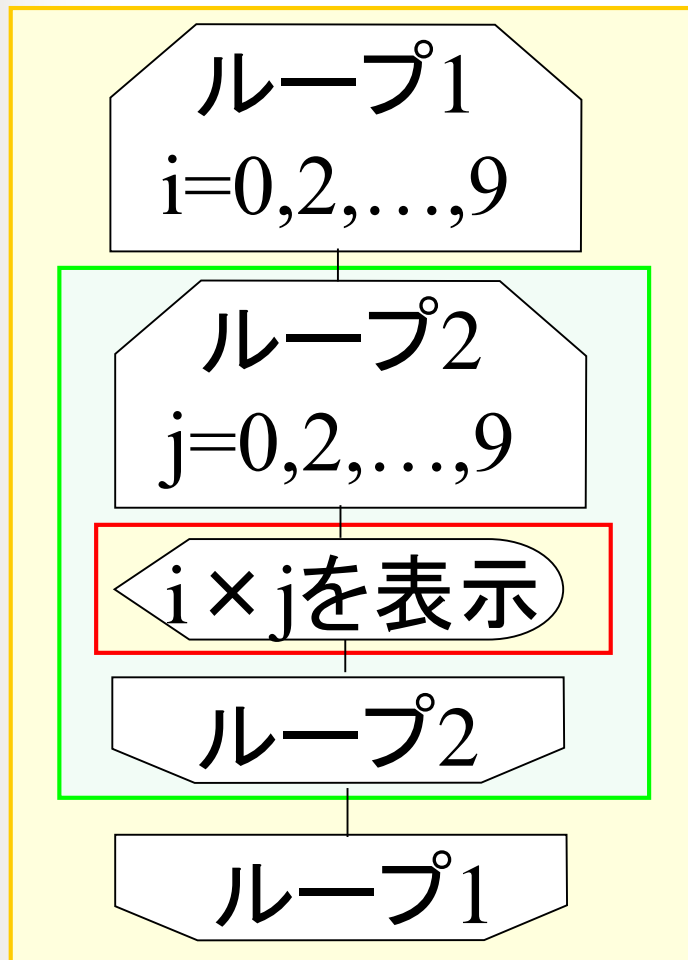




プログラミングの歴史 (2)

- E.ダイクストラらが構造化プログラミングを提唱
 - 1968年「goto文有害論」
- 構造化プログラミングに則った言語が登場
 - C, Pascal, ...
- オブジェクト指向は構造化プログラミングを含む
 - 構造化プログラミングをしっかりとマスターすること

フローチャートは構造化されていない



```
for (i = 0; i < 10 ; i++) {  
    → for (j = 0; j < 10; j++) {  
        → System.out.println(i*j);  
    }  
}
```

Java等ではインデント(字下げ)によって、見た目が構造化されている。



構造化チャート

構造化プログラミングを図で表現するために考案された。

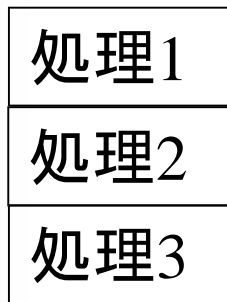
最近はあまり使われなくなってきた。

構造化チャートの種類

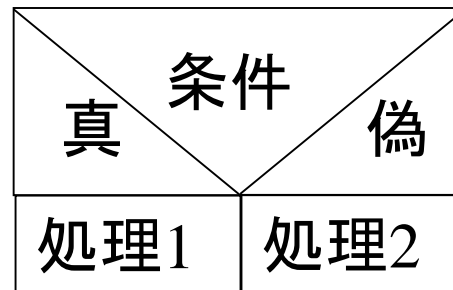
- NSチャート
- SPD
- PAD
- HCP
- YAC II
- ...

NSチャート (Nassi-Shneiderman Chart)

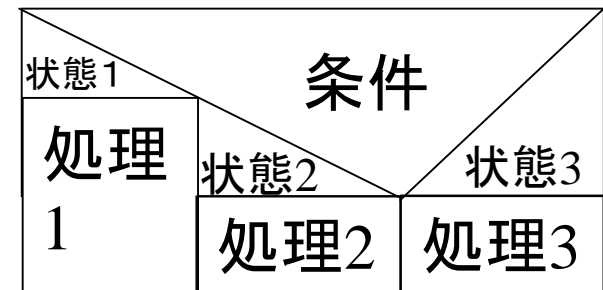
順次構造



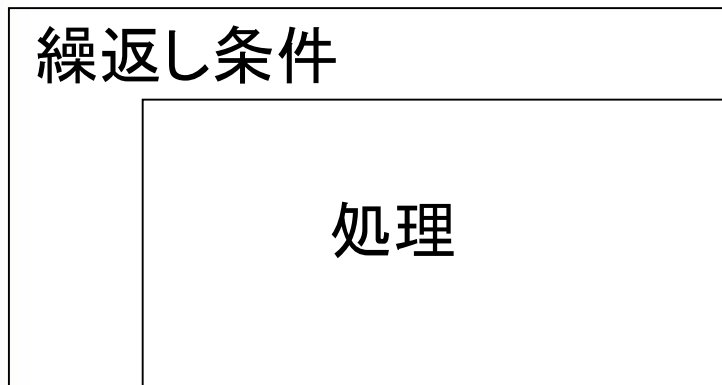
選択構造



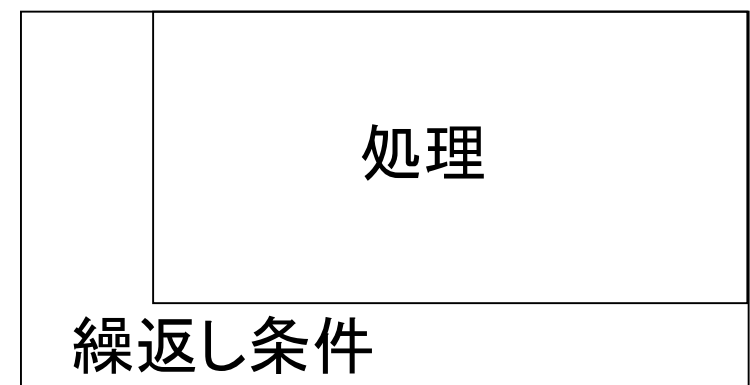
多分岐構造



繰返し構造(前判定)



繰返し構造(後判定)

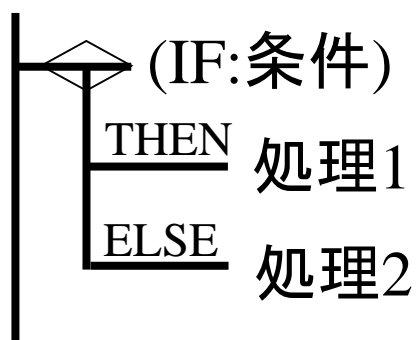


SPD (Structured Programming Diagrams)

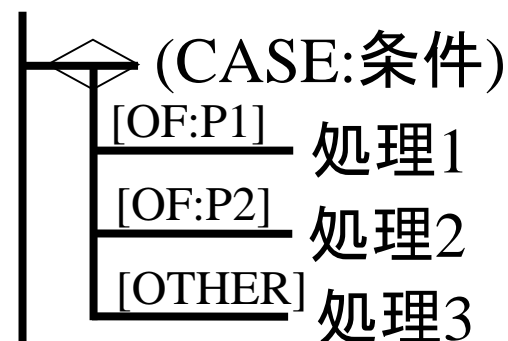
順次構造



選択構造



多分岐構造



繰返し構造(前判定)



繰返し構造(後判定)



宿題

この流れ図は最大値を表示する。
これを参考にして最小値を表示
する流れ図を作成せよ。

ただしAに入力する値は、0より
大きいものとし、0を入力するとそ
れまでに入力した値の最小値を
表示するものとする。

