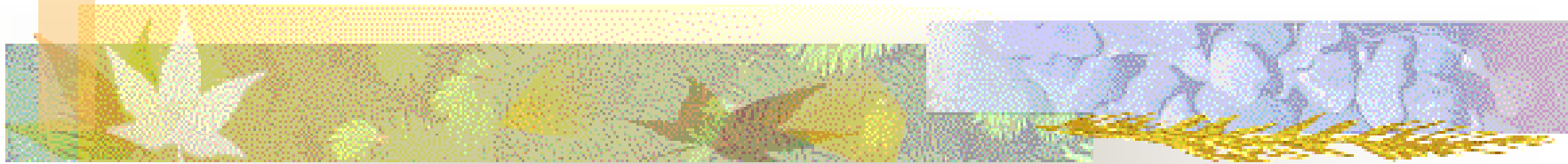


アルゴリズムとデータ構造a

4 - 擬似言語



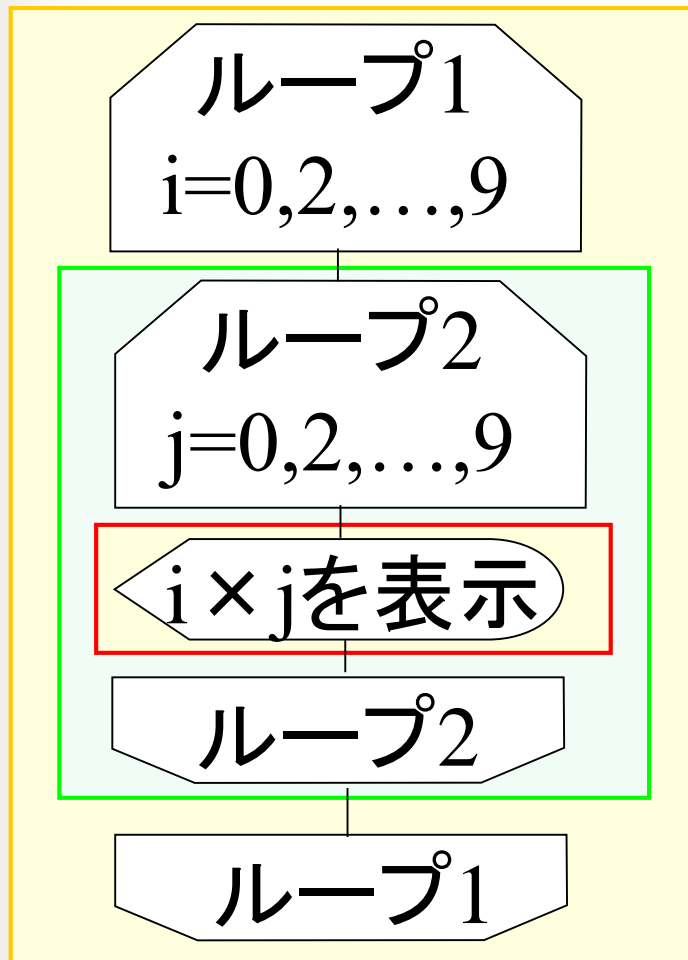
大見 嘉弘



今日の授業

- フローチャートと構造化
- 擬似言語
 - 特徴
 - 表記方法
 - 擬似言語の実例
- 簡単なアルゴリズム
 - 合計値、データの個数、平均値、最大値

フローチャートは構造化されていない

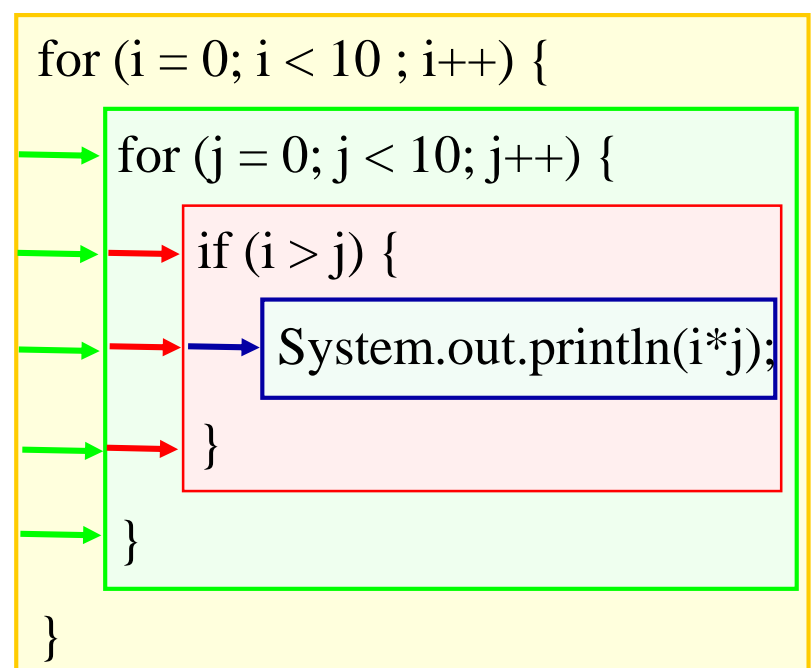
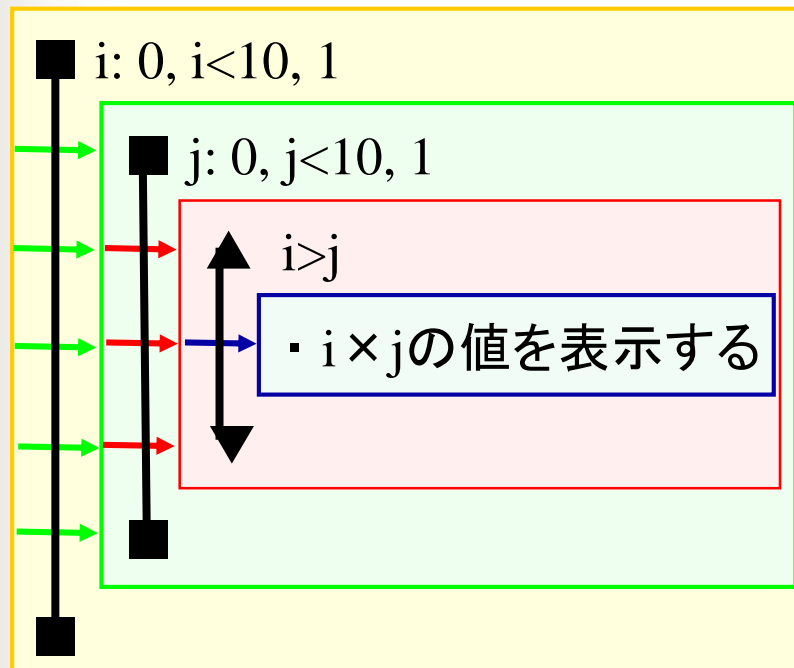


```
for (i = 0; i < 10 ; i++) {  
    → for (j = 0; j < 10; j++) {  
        → System.out.println(i*j);  
    }  
}
```

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。

擬似言語(特徴1)

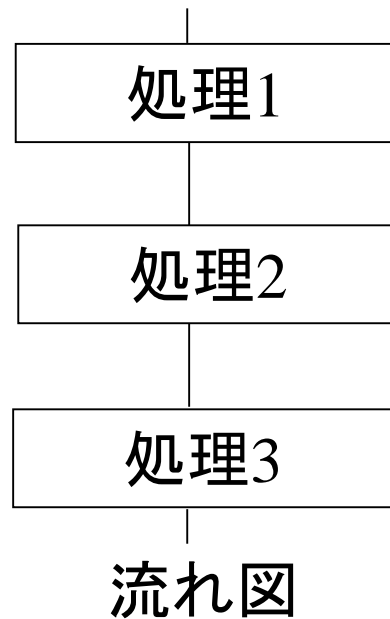
- 構造化されている(自然にインデントされる)
- Javaなどのプログラミング言語に似ている



擬似言語(特徴2)

- 情報処理試験独自の表記法
午後問題では頻繁に使用されている
- 構造化の3構造と手続き呼び出しが定められている
 - 逐次構造 (順次構造)
 - 選択構造
 - 反復構造 (繰返し構造)
 - 副プログラム(手続き、メソッド)
- 午後問題に仕様が書いてあるので変更がないか確認すること
当初は頻繁に仕様変更があった。最近はほとんど変更は起きていない。

擬似言語の表記法(逐次構造)



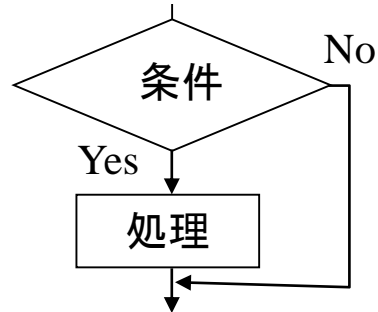
- 処理1
- 処理2
- 処理3

擬似言語

※ 行の先頭に「▪」を
忘れないように注意

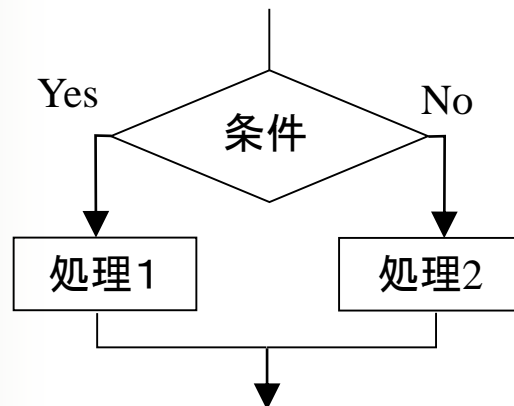
擬似言語の表記法(選択構造)

条件が成り立つとき処理(成り立たないときは何もしない)



↑ 条件
↓
・ 処理

条件が成り立つとき処理1、成り立たないとき処理2



↑ 条件
↓
・ 処理1

・ 処理2

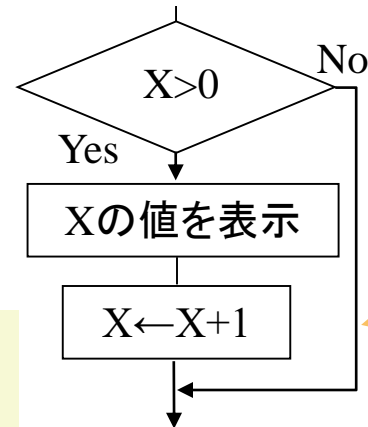
選択構造の例

$X > 0$

- ・ X の値を表示
- ・ $X \leftarrow X+1$

逐次
構造

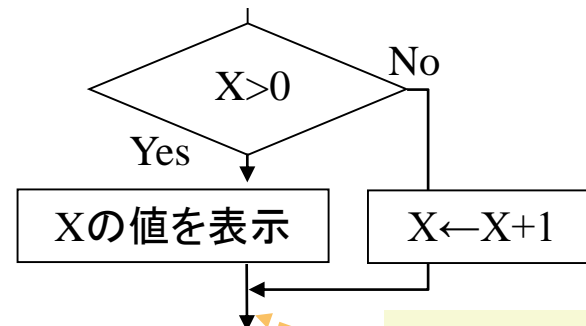
選択構造の中に逐次
構造が含まれている



$X > 0$

- ・ X の値を表示

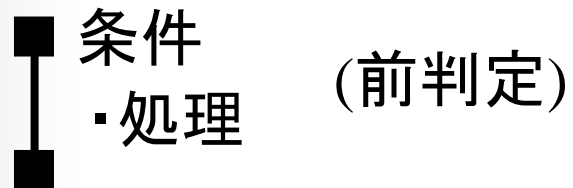
- ・ $X \leftarrow X+1$



出口は必ず1つ

擬似言語の表記法(反復構造)

条件が真の間、処理を繰返す

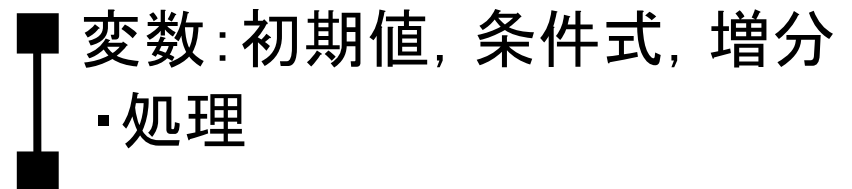


Javaのwhile文と同様



Javaのdo～while文と同様

Javaのfor文と同様



```
for (変数 = 初期値;条件式;増分の式) {  
    処理;  
}
```

反復構造の例(1)

- ・ Aに入力

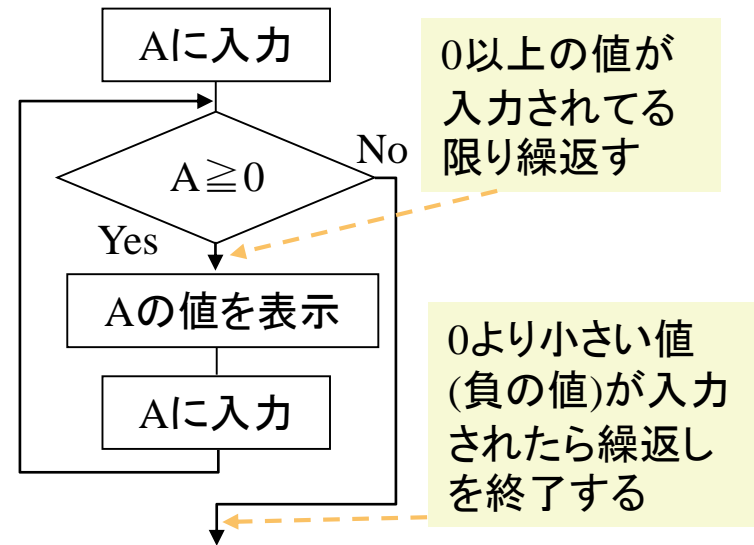
■ $A \geq 0$

- ・ Aの値を表示
- ・ Aに入力

↑
逐次
構造

反復構造の中に逐次
構造が含まれている

=

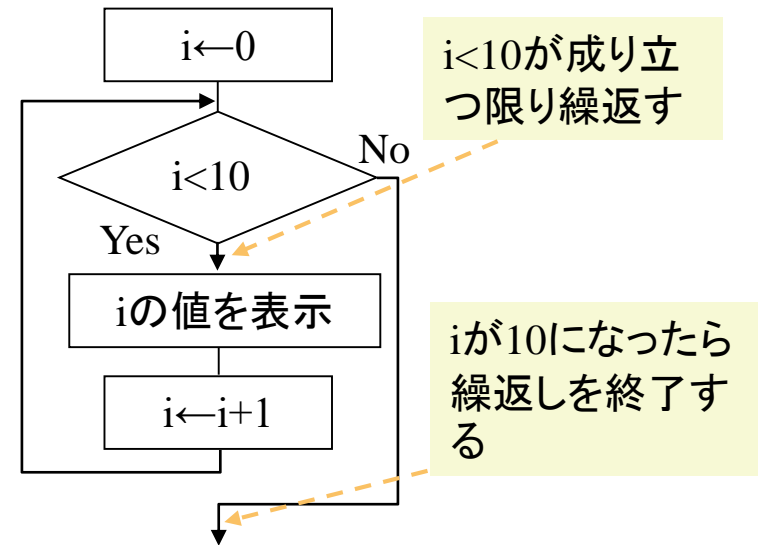
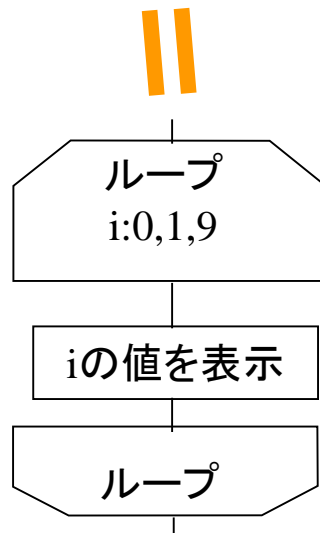
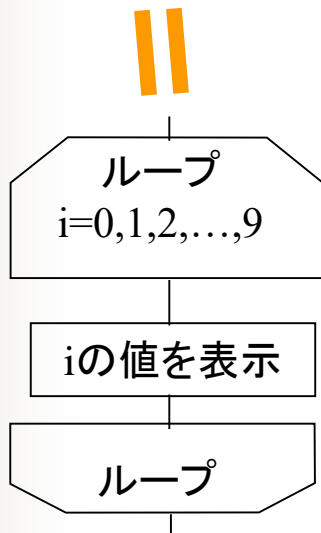


反復構造の例(2)

■ 変数: 初期値, 条件式, 増分
■ 処理

■ $i: 0, i < 10, 1$
■ i の値を表示

0,1,2...9 を表示する(つまり10回繰返す)



ループ端を使用した場合

擬似言語の実例

(平成14年度秋期午後問題より抜粋)

宣言部

○ 副プログラム名: `GetLine(Text[], N, Sp, Moji, Ep)`
○ 文字型: `Text[]`
○ 整数型: `N, Sp, Moji, Ep`
○ 論理型: `Loop`

• `Ep ← Sp`

• `Loop ← true`

■ `Ep < N and Loop`
▲ `Text[Ep] ≠ "." and`
• `Ep ← Ep + 1`
• `Loop ← false`
▼

b

処理部

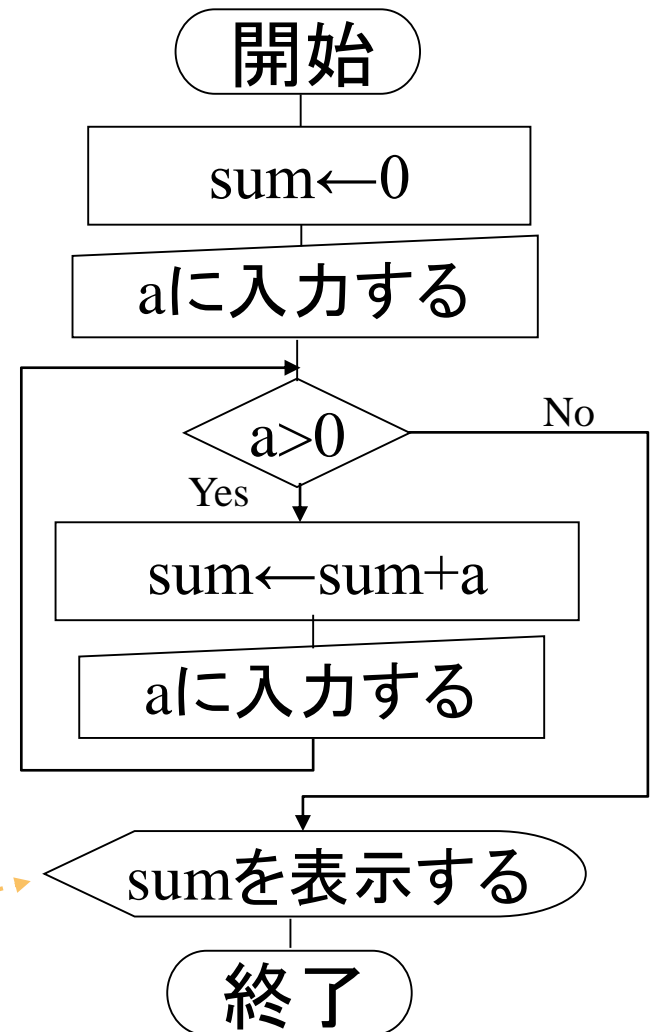
合計値を求める

○整数型: sum, a

- $\text{sum} \leftarrow 0$
- aに入力する
- $a > 0$
 - $\text{sum} \leftarrow \text{sum} + a$
 - aに入力する
- sumを表示する

=

0以下の値を入力したら
繰返しを終了し、合計値
(sum)を表示する



データの個数を求める

(何個入力があったか数える)

○整数型: count, a

▪ count ← 0

▪ aに

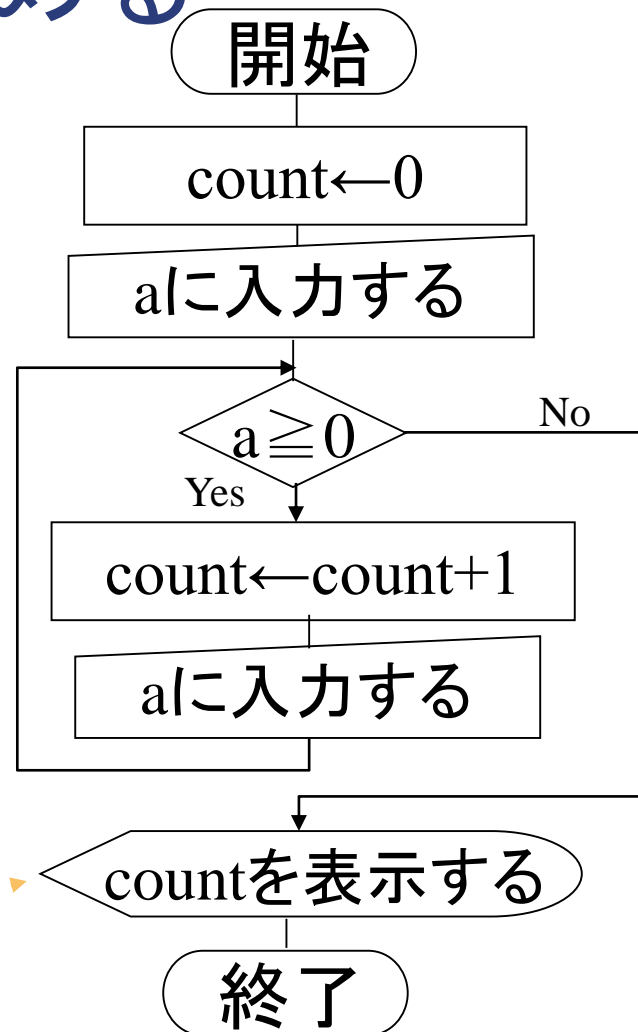
■ $a \geq 0$

▪ count ← count + 1

▪ aに

▪ countを表示する

負の値を入力したら繰返しを終了し、データの個数 (count) を表示する



平均値を求める

○整数型: sum, count, a

○実数型: ave, rsum

- $sum \leftarrow 0$

- $count \leftarrow 0$

- aに入力する

- $a \geq 0$

- $sum \leftarrow sum + a$

- $count \leftarrow count + 1$

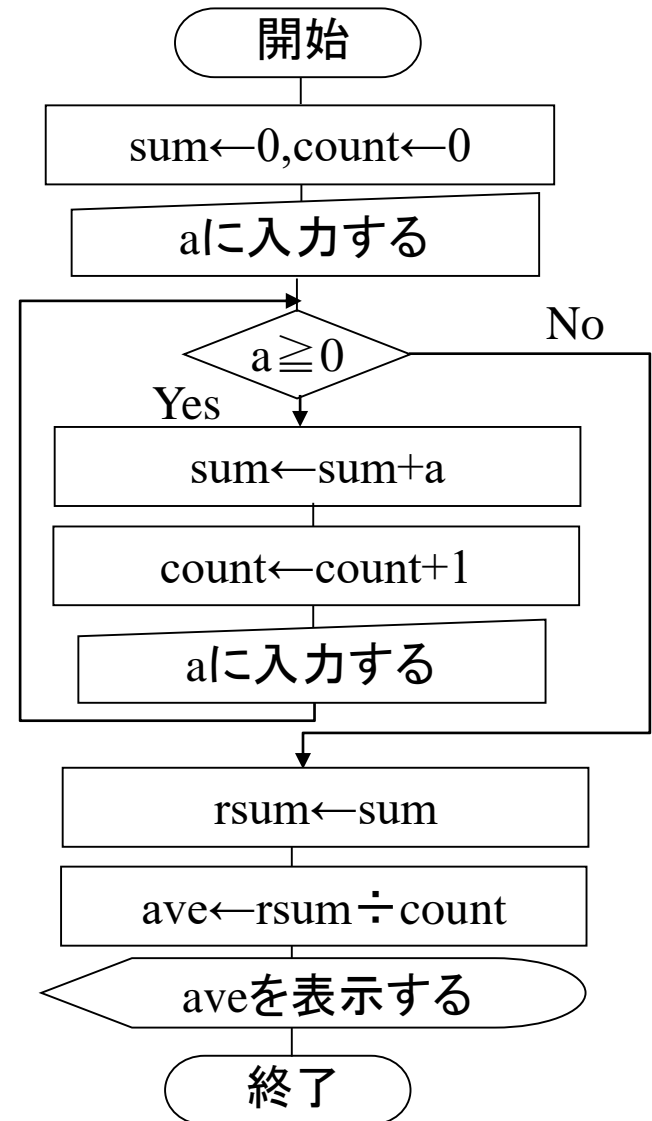
- aに入力する

- $rsum \leftarrow sum$

- $ave \leftarrow rsum \div count$

- aveを表示する

=



最大値を求める

○整数型: max, a

- ・ aに入力する
- ・ $\text{max} \leftarrow a$

■ $a \geq 0$

↑ $\text{max} < a$

- ・ $\text{max} \leftarrow a$

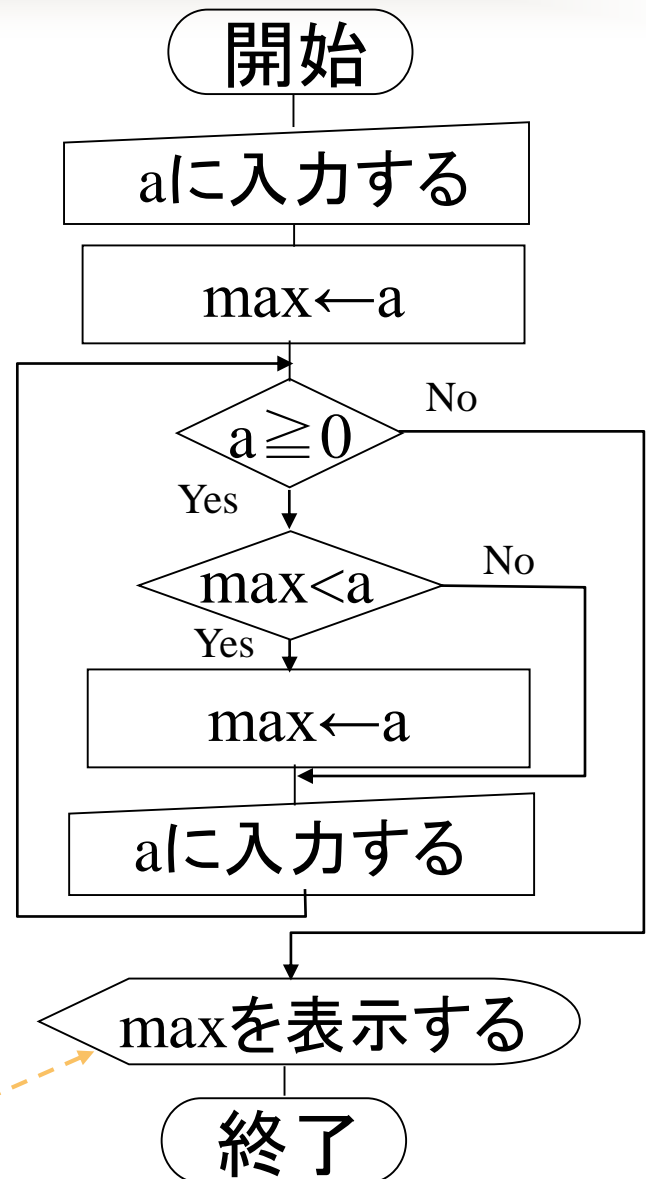
↓
・ aに入力する

- ・ maxを表示する

負の値を入力したら繰返しを終了し、最大値(max)を表示する

繰返しの途中では、その時点での最大値がmaxに入っている

現時点の最大値(max)より今度の入力値aのほうが大きい時



擬似言語(副プログラム)

■ Javaのメソッドに相当する

○プログラム名:合計処理

○整数型: $T[] = (2, 8, 15, 4)$

○整数型: N, ans

▪ $N \leftarrow 6$

▪ $SUM(T, N, ans)$

戻り値も(...)の中を書く
ことに注意。

引数
 T, N

○副プログラム名:

$SUM(T[], N, ans)$

○整数型: $T[], N, ans, i$

▪ $ans \leftarrow 0$

▪ $i \leftarrow 1$

■ $i \leq N$

▪ $ans \leftarrow T[i] + ans$

▪ $i \leftarrow i + 1$

■

戻り値
 ans

宿題

右の流れ図(最小値を表示するプログラム)の内容を、擬似言語で描け。なお、入力する値は実数値とする。

