

# 情報処理工学 第11回

藤田 一寿

公立小松大学保健医療学部臨床工学科

# プログラミングの基礎 フローチャート

# ■ ソフトウェアを作る上で必要となる事柄

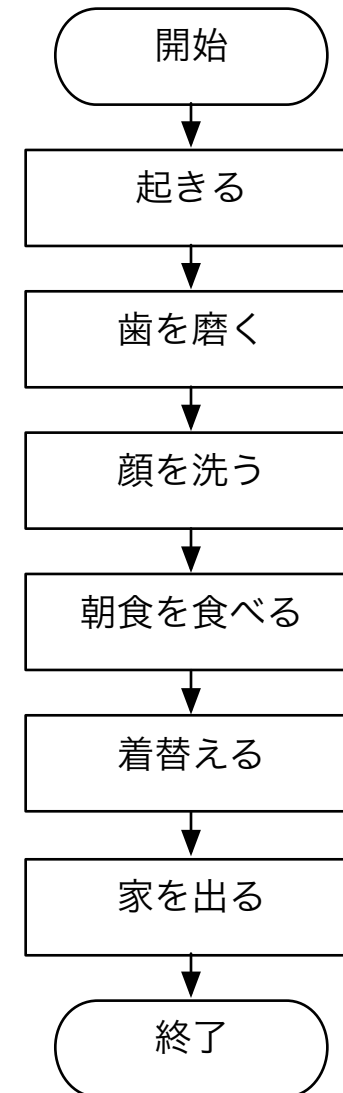
- アルゴリズム
  - 計算のやり方・手順
  - 計算と書いているが、情報の世界ではソフトウェアにやらせたいこと（実現したいこと）とだと思っていれば良い.
- プログラミング言語
  - ソフトウェアを作るための人工言語. アルゴリズムに基づいてプログラミング言語を用いソフトウェアを作成.

# ■ フローチャート

- コンピュータはソフトウェアがないと動かない.
- ソフトウェアは正しい処理手順（アルゴリズム）で作れていないと動かない.
- 処理手順を図で分かりやすく描く方法の一つがフローチャート.
- フローチャートはJIS規格により仕様が定められている.
- フローチャートは、処理手順を分かりやすく記述できるため、ソフト開発以外の場面でもよく用いる（ソフト開発以外で見るほうが多いかも）.
  - 事務手続きの方法
  - 緊急事態への対応手順
- 国家試験のためには少なくともフローチャートを読めるようにならないといけない.

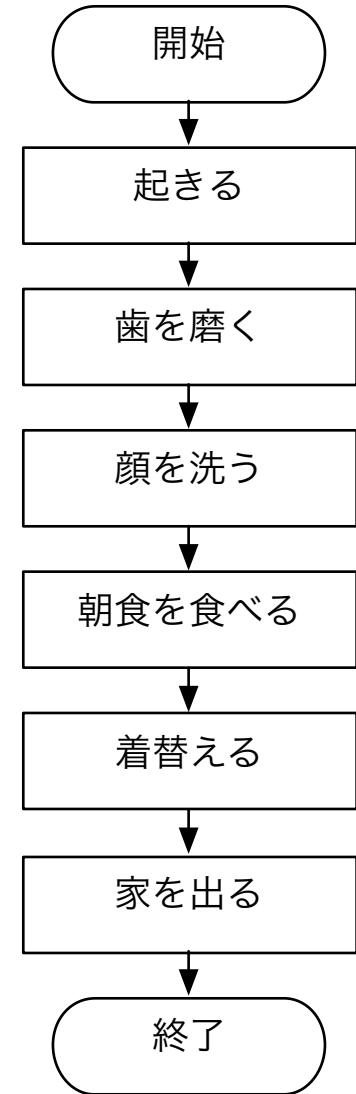
## ■ 例：朝起きてから家を出るまで

1. 起きる
2. 歯を磨く
3. 顔を洗う
4. 朝食を取る
5. 着替える
6. 家を出る



## ■ 逐次処理

- 上から下に順番に1つずつ処理を行う.
- 必ず処理は順番通り行う.
- 勝手に順番を抜かしたり, 戻ったりしてはいけない.
- 処理のはじめと終わりは角が丸い長方形? で描く.
- 処理は長方形で描く.
- 処理は上から下へ描く.
- 処理は線または矢印でつなぐ.



## ■ 例：家を出てから大学に着くまで

1. 家を出る
2. 弁当を買いたいなら3へ, 買わないなら5へ
3. セブンイレブンへ行く
4. 弁当を買う
5. 大学に到着

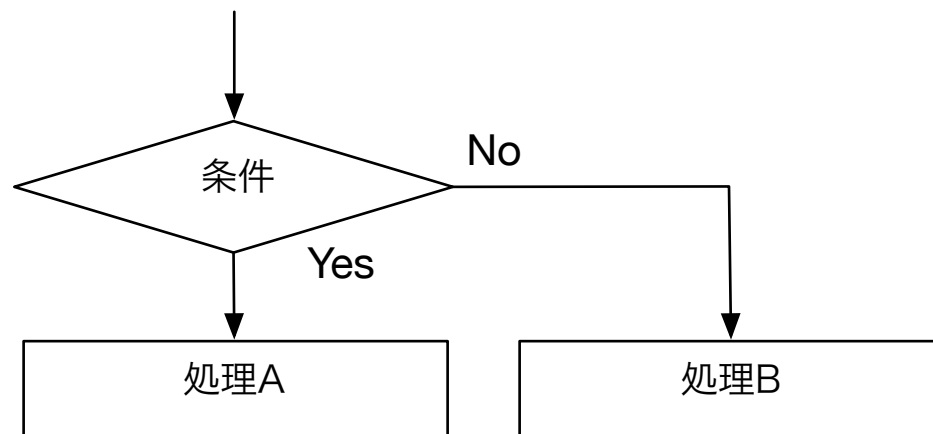
条件によって処理が分岐するような  
処理はどうすればよいか？

条件分岐

# 条件分岐

- 条件によって処理の流れが変わること.
  - 条件によって処理が分かれる様子を分岐とよんでいる.
- フローチャートでは条件分岐はひし形で描く.
- 条件を満たした場合の処理の流れにはYes, 条件を満たさない場合の処理の流れにはNoを描く.

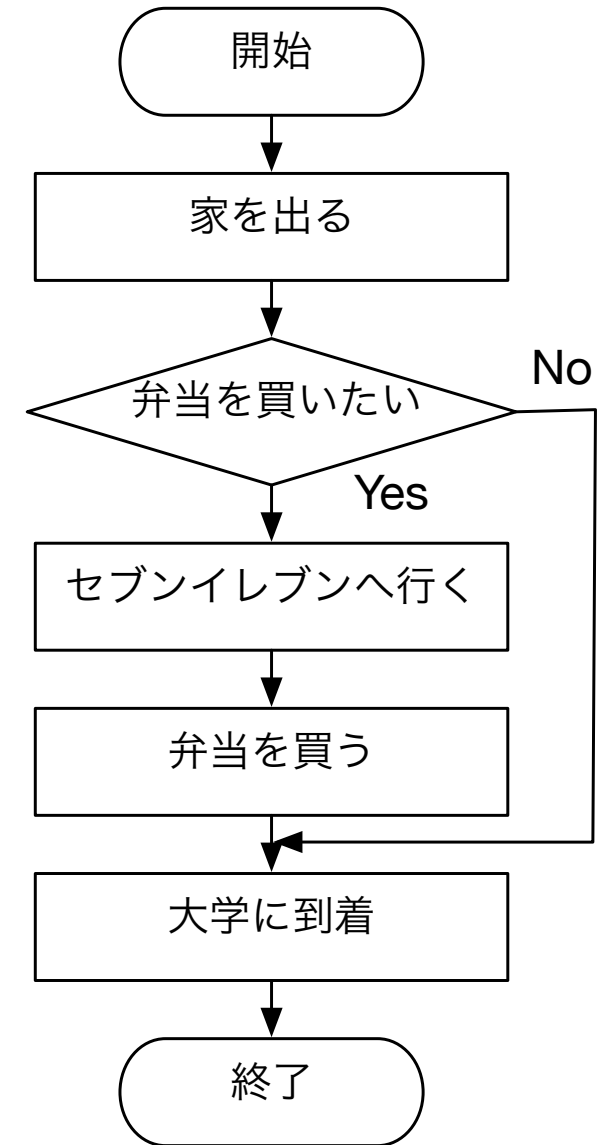
ひし形の中に書かれた条件を満たせば, Yesと書かれた矢印に進み処理Aを実行する.  
条件を満たさなければ, Noと書かれた矢印に進み処理Bを実行する





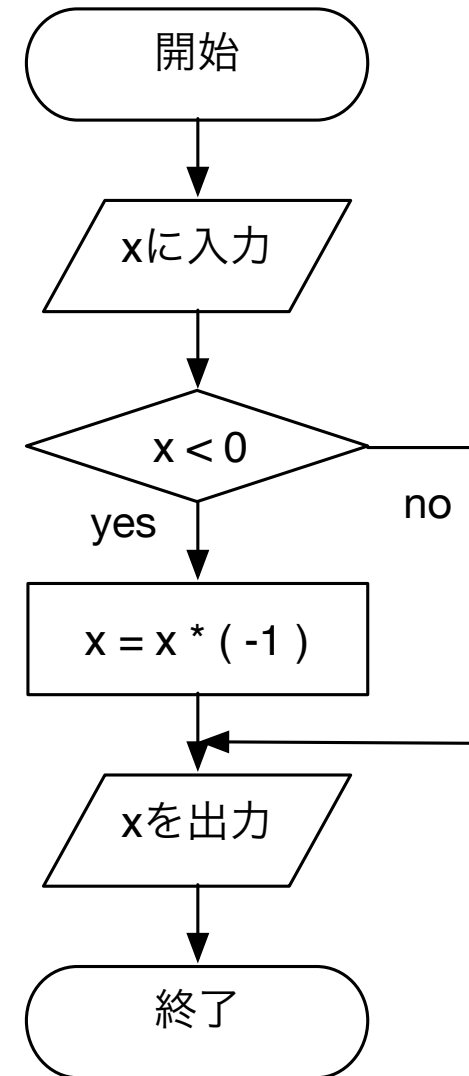
## ■ 例：家を出てから大学に着くまで

1. 家を出る
2. 弁当を買いたいなら3へ, 買わないなら5へ
3. セブンイレブンへ行く
4. 弁当を買う
5. 大学に到着



## ■ 例：絶対値を求めるためのフローチャート

- 入力した数値の絶対値を表示する.
1. xに数値を代入する.
  2. xが負なら3へ, そうでなければ4へ
  3. xに-1を掛ける.
  4. xを出力する



\*は掛け算を表す.

## ■ フローチャートで用いられるプログラミング言語的表現

- 臨床工学士の国家試験では、フローチャートの問題でプログラミングで用いられる記述方法が用いられる.
- 変数：アルファベットで書かれる。数値や文字を入れる箱と思えば良い.
- 数値の代入：数値の代入は=もしくは、 $\leftarrow$ で書かれる。
  - 例：xに0を代入  $x = 0$ もしくは  $x \leftarrow 0$
- 不等号：条件分岐で用いられる。ただし次の表現には注意。
  - xは0以上  $x \geq 0$
  - xは0以下  $x \leq 0$
  - xは0と等しい  $x == 0$
- 国家試験の中で表現が統一されていないので、雰囲気で察することも必要.

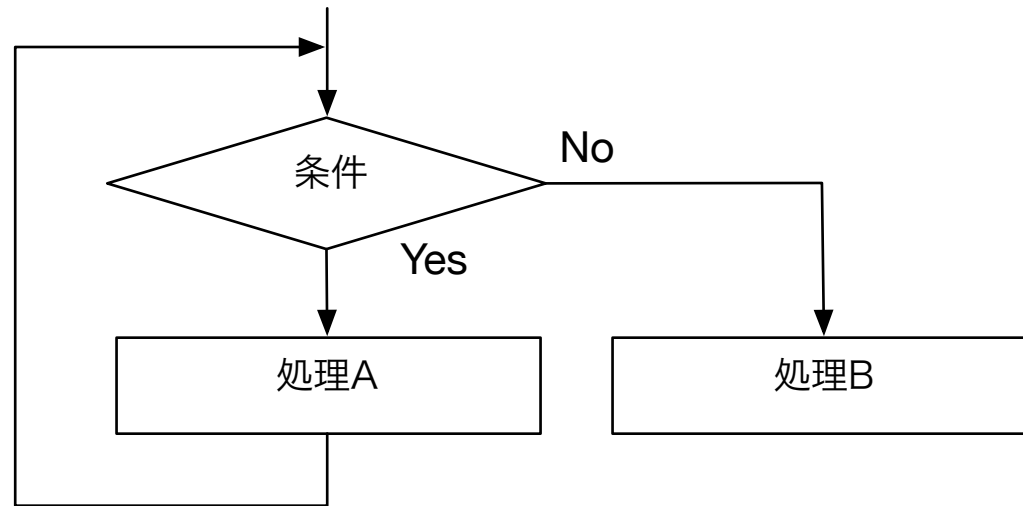
## ■ 演習

- 成績を入力し，60点以上なら「合格」，60点未満なら「不合格」を出力するフローチャートをかけ．ただし，入力を入れる変数はxとせよ．

## ■ 繰り返し処理（ループ処理）

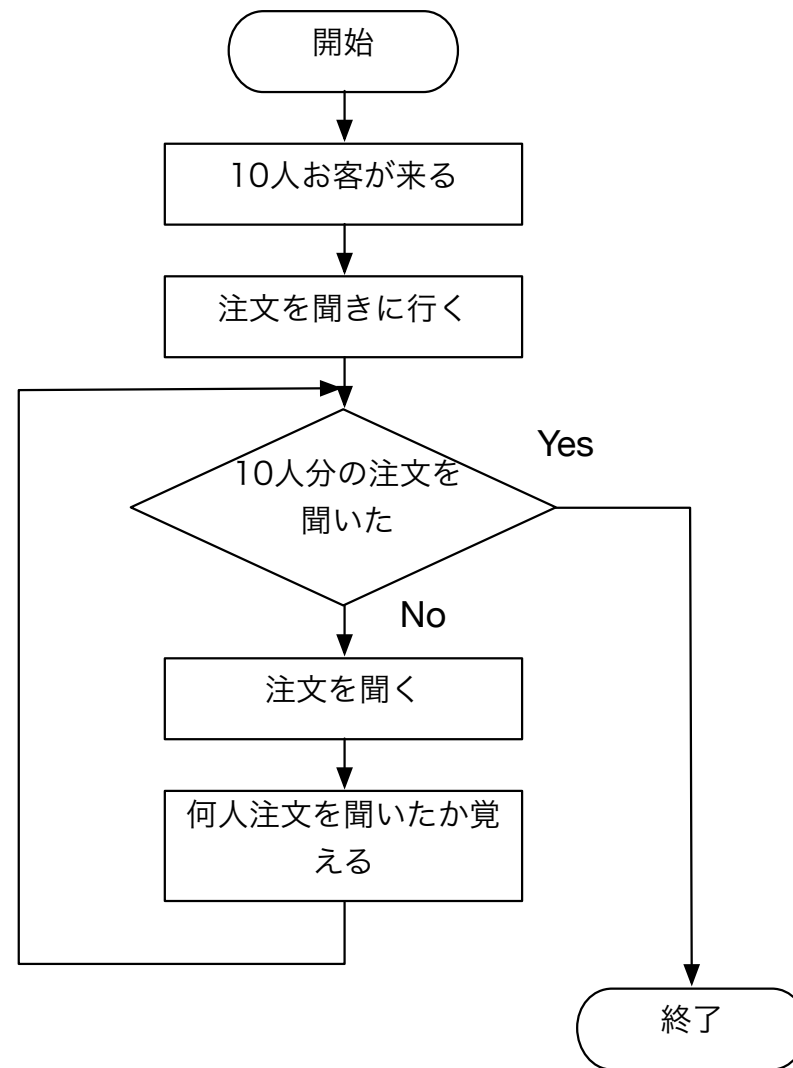
- ある処理を何度も繰り返すことを，繰り返し処理（ループ処理）と言う．

処理に戻る構造がある



# 10人のお客さんの注文を聞く

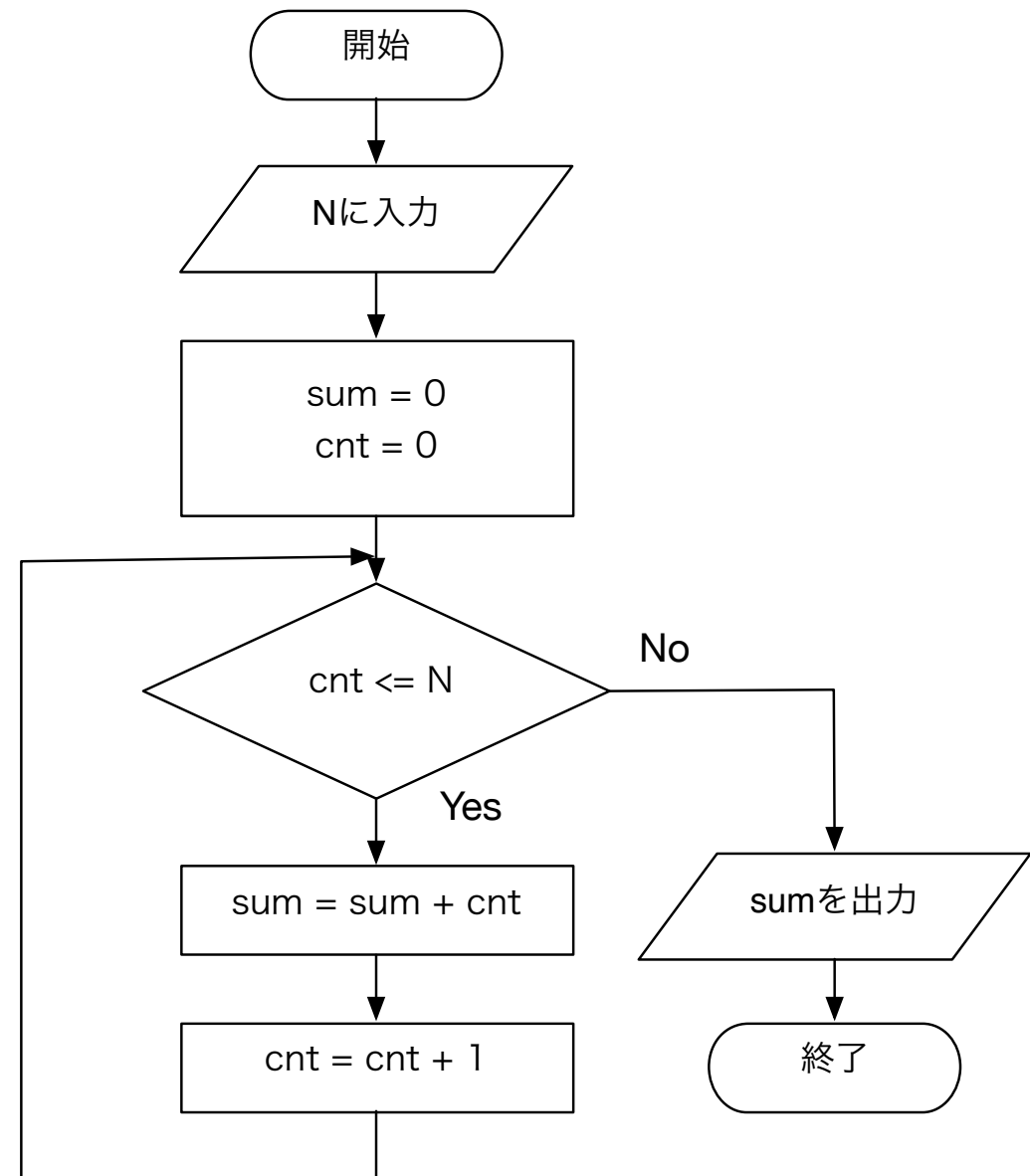
1. 10人お客さんが来る.
2. 注文を聞きに行く.
3. 10人目まで注文を聞いていなければ4へ, 聞いていれば7へ
4. 注文を聞く.
5. 何人の注文か覚える.
6. 3に戻る
7. 終了



## ■ 例：Nまでの自然数の和を出力する.

1. 入力をNに入れる.
2. sumに0を入れる.
3. cntに0を入れる.
4. cntがN以下であれば5へ,  
そうでなければ8へ
5.  $\text{sum} = \text{sum} + \text{cnt}$
6.  $\text{cnt} = \text{cnt} + 1$
7. 4に戻る
8. sumを出力

- sumに和が入る
- cntはカウンタと呼ばれる。足した回数を保存する。自然数の和なので、足した回数は足す数と同じ。
- =は代入を意味する。



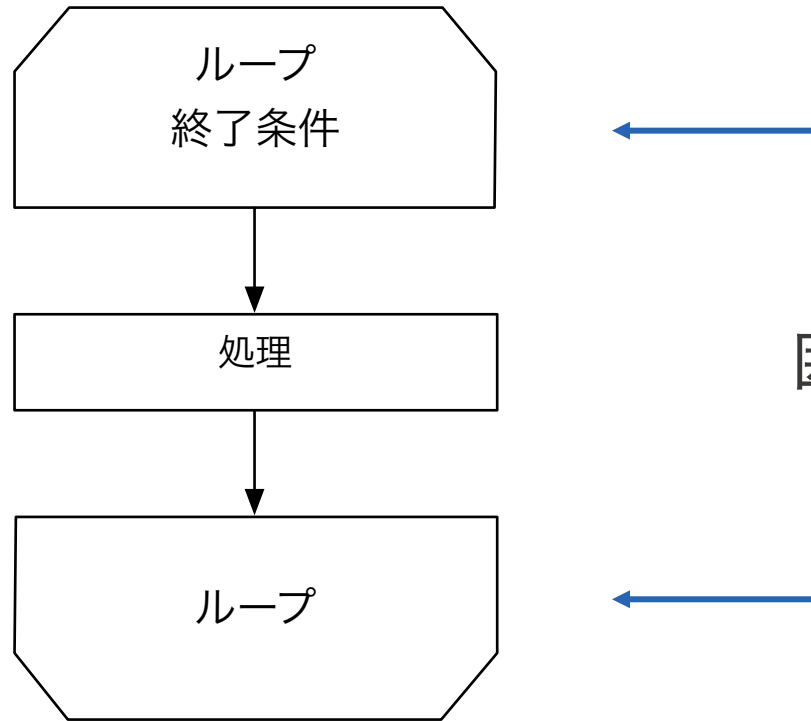
## ■ 演習

- 1以上入力Nまでの整数の中で、偶数のみを出力する処理のフローチャートをかけ。ただし、Nは1以上の整数とする。また、 $a \div b$ の余りは $a \% b$ で計算されたとする。



## ■ ループ処理のための特別な図形

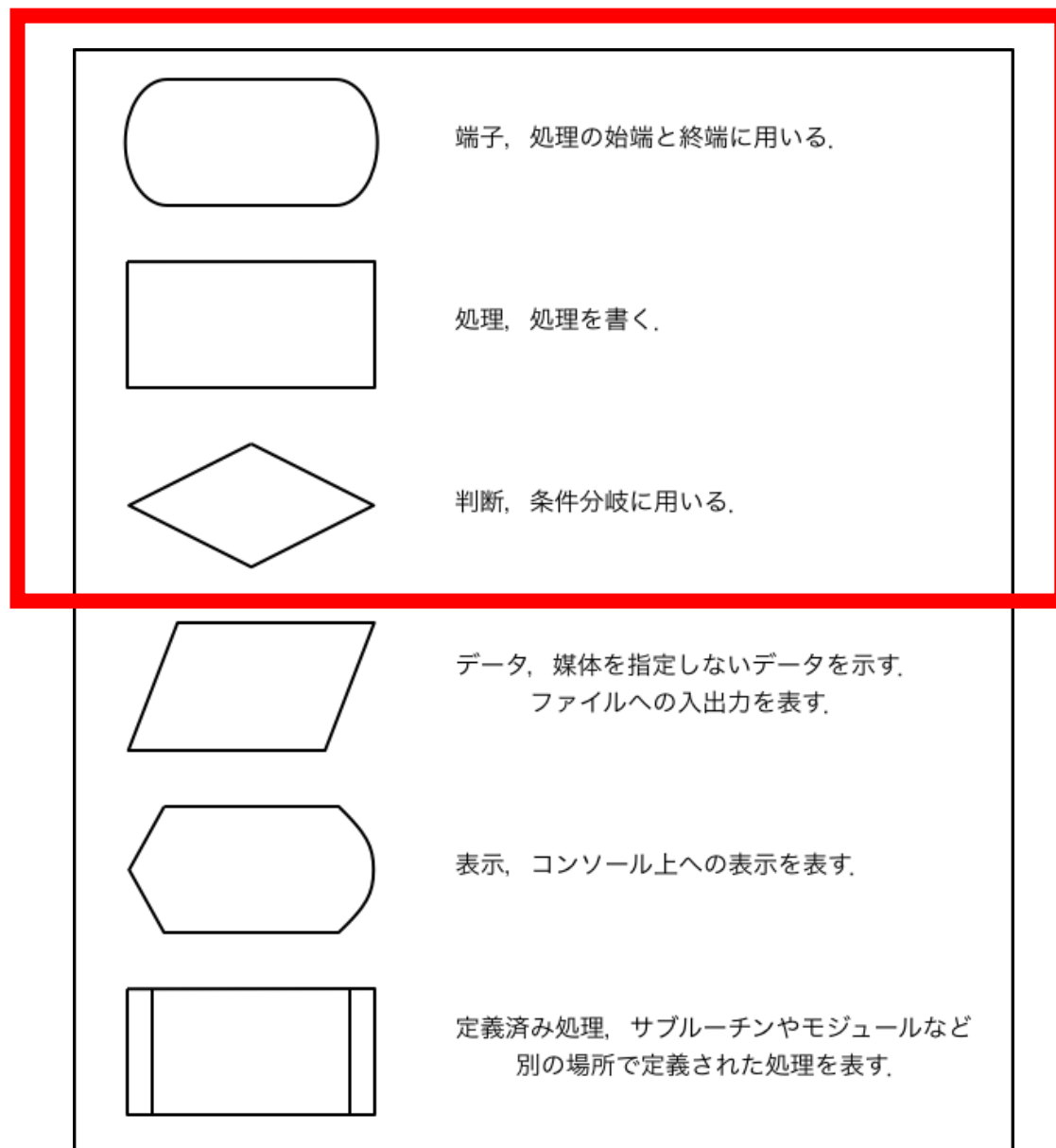
- 条件分岐により繰り返し処理を記述したが、繰り返し処理専用の図形も存在する。



囲まれば部分が繰り返す。

# ■ フローチャートの構成要素のまとめ

必ず覚える



これ以外の図形が出てきても、  
図形の中に処理が必ず書かれて  
いるので、それ読めばフロー  
チャートは理解できる。

## 演習

- 図のフローチャートに基づいて作成されたプログラムを実行した結果, 出力されるZはいくらか.  
(第29回国家試験改)

