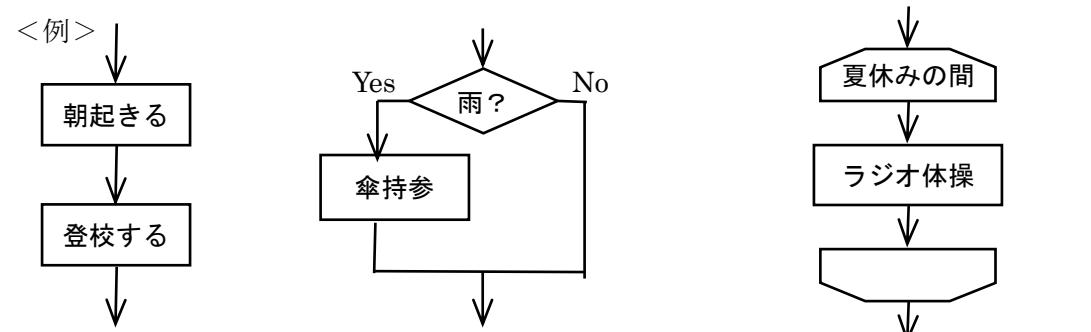
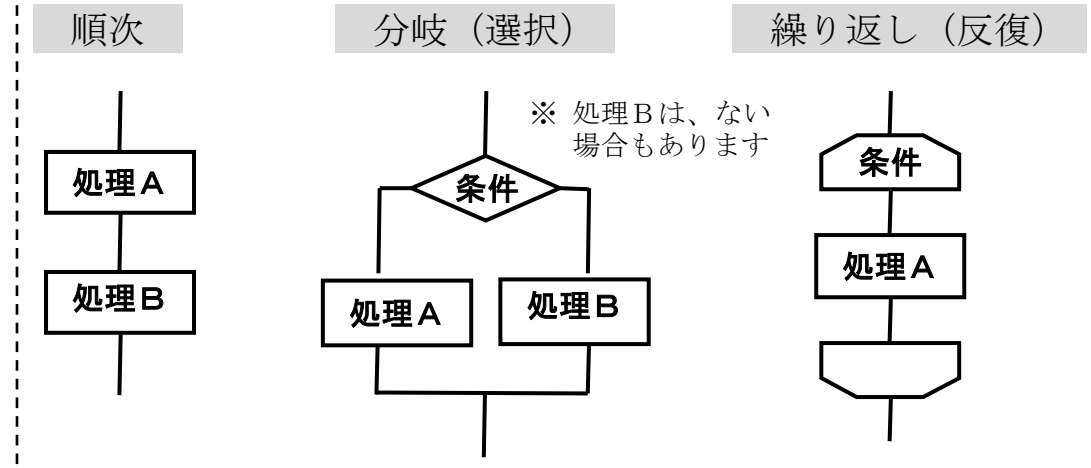
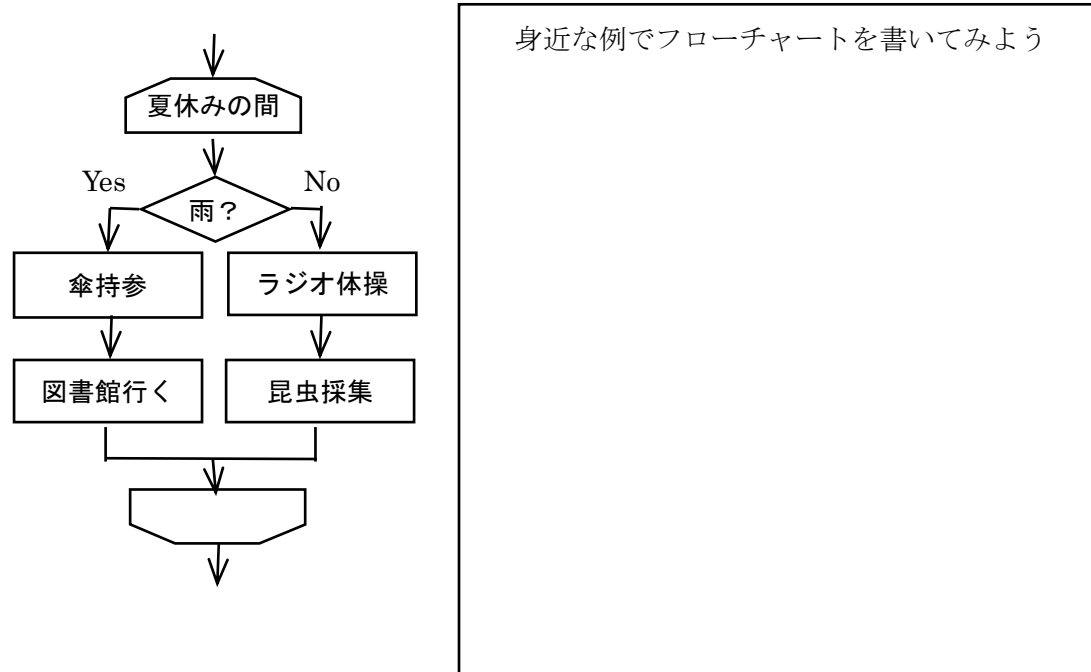


制御構造の基本要素 (フローチャート)



上記三つの要素を組み合わせることで、さまざまな処理の流れが表せます。



身近な例でフローチャートを書いてみよう

ツール PenFlowchart について

直感的な操作でフローチャートを作成し、実際に計算や制御を行うことができます。

① パーツ
② 変数の宣言
③ フローチャート作成エリア

④ 実行ボタン等
⑤ 状態 (入力待ち/実行終了等)
⑥ 実行画面

※実行ウィンドウの白い部分は今回は使いません (Pen プログラム編集エリア)

パーツの種類

←	・・・代入	条件	・・・分岐
入力	・・・キーボードから値を入力	条件	・・・繰り返し
出力	・・・画面に値を表示		

【配置】ドラッグ&ドロップ 【編集】パーツをダブルクリック

【削除】パーツ右クリックで表示されるメニューから

【注意】実際のプログラミングをする場合の注意としてもほぼ当てはまります。

- 変数や数字、マイナス記号などはすべて半角 (日本語表示する文以外すべて)
- 「× (乗算)」は「* (アスタリスク)」、「÷ (除算)」は「/ (スラッシュ)」
- 「←」は代入する、という意味。「x ← 100」の場合、「x に 100 を代入する」という意味 (実際のプログラミング言語では「=」を使うことが多い)
- 文字列はかぎ括弧でくくる (実際のプログラミング言語ではダブルクォーテーション' やシングルクォーテーション' を使うことが多い)
- 変数は必ず図中②で型宣言 (整数・文字列等) をする。型宣言がないと右図のようなエラーが表示される →

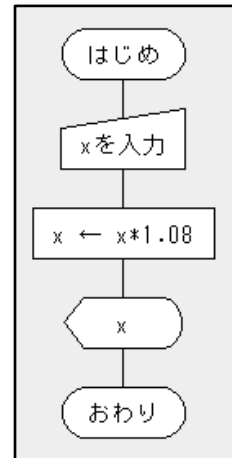
社会と情報 [発展] 制御構造とフローチャート No. 2

年 組 番

ツール PenFlowchart を使って処理の流れを表し、実行してみよう。

順次処理

<実習 1-1> 左下のフローチャートは「消費税込み金額を求める処理」です。PenFlowchart を使って作成し、実行ウィンドウで実行してみましょう。入力待ちになるので、キーボードから値を入力して画面に結果が表示されることを確認しよう。



整数	x
実数	
文字列	
真偽	

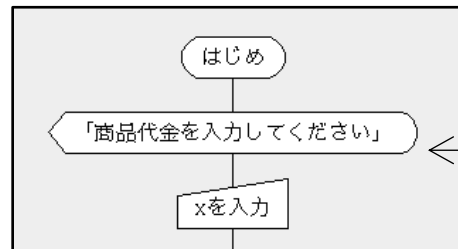
変数の型宣言を忘れずに！

再度実行するには「始めから実行」ボタンを押します（繰り返し処理を作成していないので、一度の実行で一度だけしか計算できません）。

<問> 500 を入力した場合の結果表示は？

(答) 540

<実習 1-2> 左下のフローチャートは、実習 1-1 のフローチャートの最初の部分を使いやすく改造したものです。これを更に改造してもっと使いやすくしてみよう。例えば、「消費税込み金額は〇〇円です。」と表示するなど。

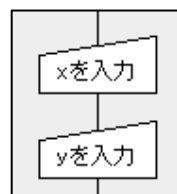


出力

画面表示なので出力パーツを使います。「入力してください」という文章にだまされないように！

<応用実習 1-3> 商品代金 (変数 x) と消費税率 (変数 y) をキーボードから入力し、消費税込み金額を画面に表示するフローチャートを作ってみよう。

※変数の入力是一个ずつ



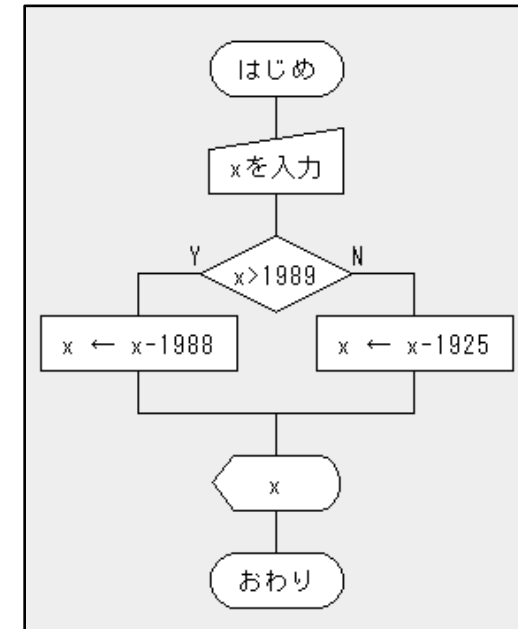
※型宣言はコンマ区切りで

整数	x, y
実数	
文字列	
真偽	

ヒント : $x \leftarrow x + x*y/100$ など

分岐処理

<実習 2-1> 次のフローチャートは「西暦を和暦に変換する処理」です。キーボードから、いろいろな値を入力して結果表示を確認しよう。



<問> 2017 を入力した場合の結果表示は？

(答) 29

<実習 2-2> 実習 2-1 のフローチャートを使いやすく改造してみよう。

- 例
- ・「西暦を入力してください」とメッセージを表示
 - ・「平成」「昭和」を表示する など

<実習 2-3> 明治と大正も表示できるよう、実習 2-2 のフローチャートを改造しよう。

明治 : 1868 年～1912 年

大正 : 1912 年～1926 年

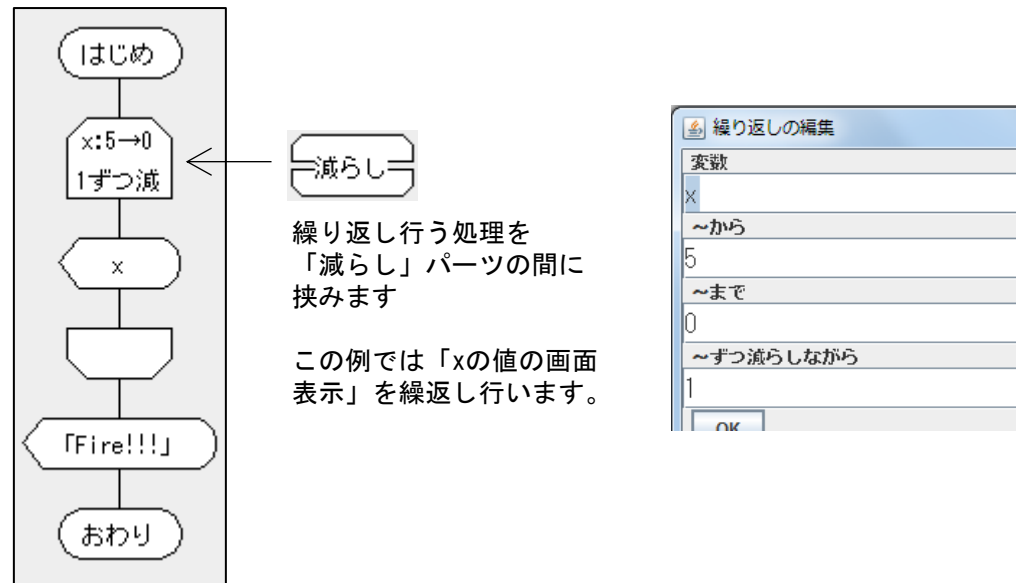
昭和 : 1926 年～1989 年

平成 : 1989 年～現在

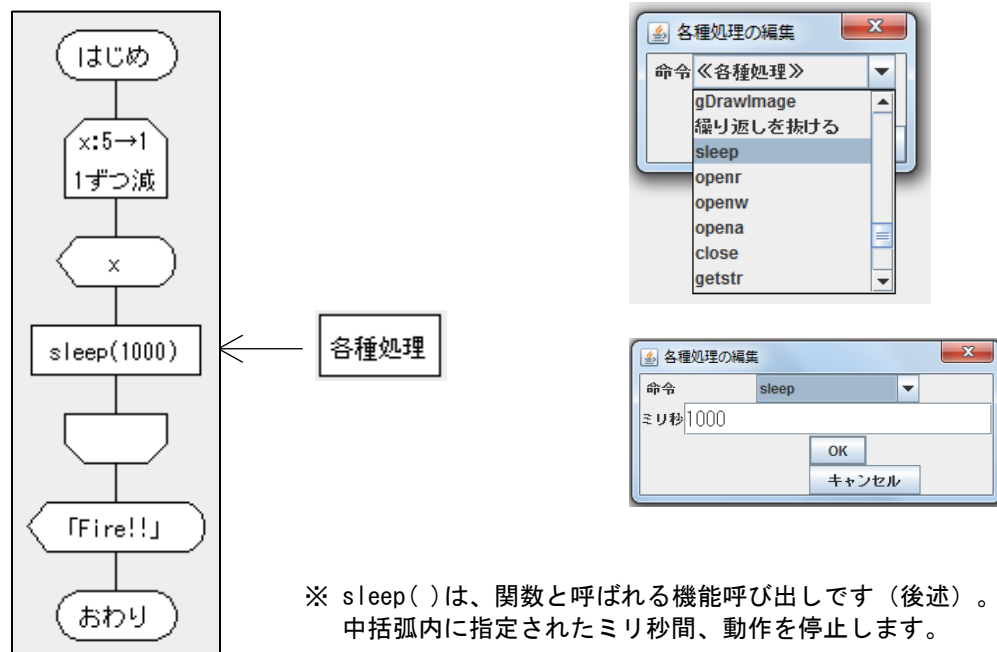
<考えよう 2-4> 正しくない数字 (1868 年以前、2018 年以降) が入力されたらエラーメッセージを表示するためには、どうすればよいだろうか？また、2 年の途中から新しい元号が変わることに対応するには、どうすればよいだろうか？

繰り返し処理

<実習 3-1> 左下のフローチャートは「ロケット発射のカウントダウンを行う処理」です。変数 x の値を 5 から 1 ずつ減らしながら画面表示します。実行ウィンドウで実行してみよう。



<実習 3-2> カウントダウンが早すぎるので、「各種処理」パーツから「sleep」を入れてみよう。「sleep」を使うと動作が一時停止します。1000 ミリ秒（つまり 1 秒）指定することで、1 秒ずつカウントダウンできるようになります。

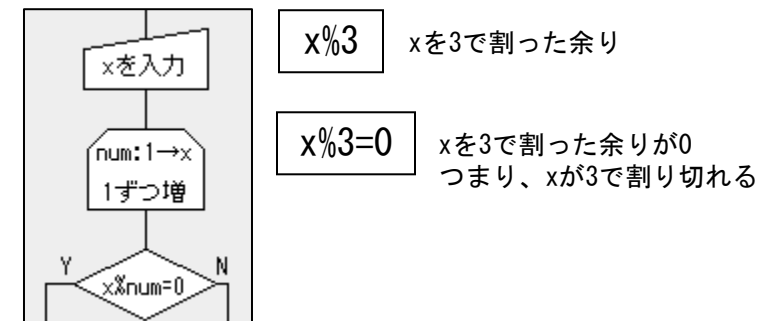


※ sleep() は、関数と呼ばれる機能呼び出しです（後述）。中括弧内に指定されたミリ秒間、動作を停止します。

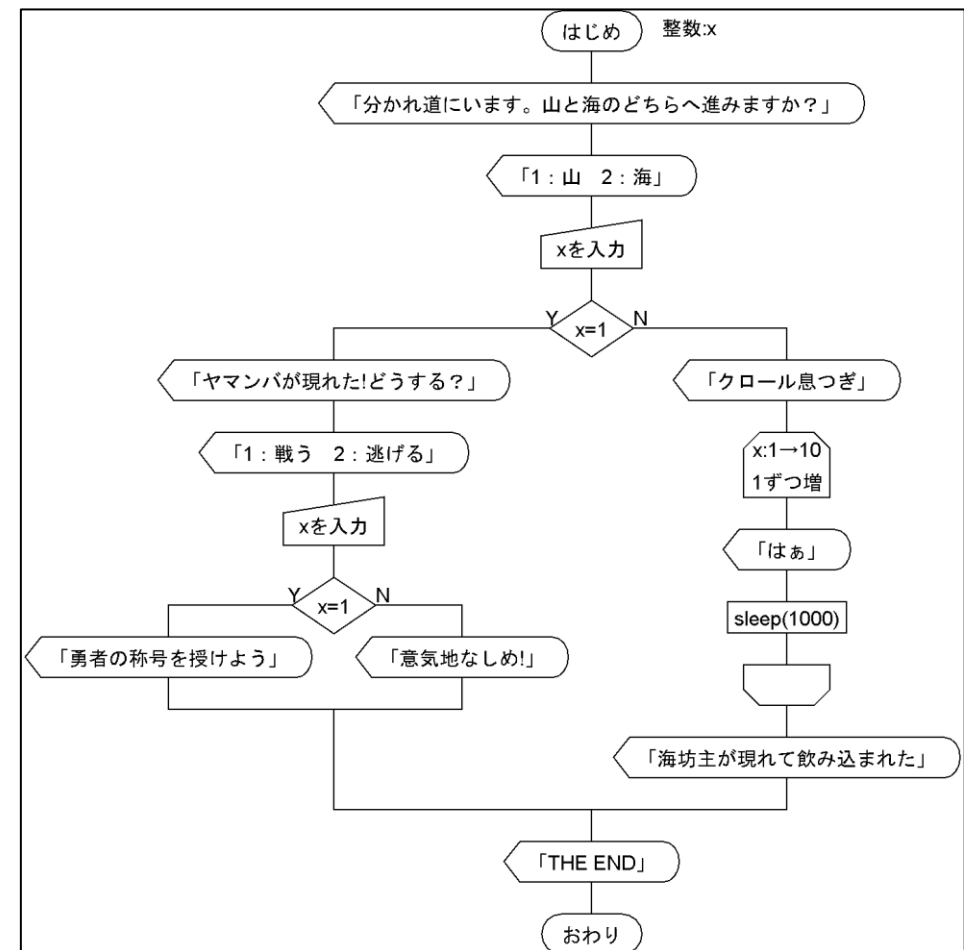
<考えよう 3-3> 繰り返し処理を使わずに作ることはできるだろうか？繰り返し処理を使う場合のメリットは何？

応用（順次・分岐・繰り返しの組み合わせ）

<実習 4-1：応用> 入力した整数の約数を全て表示するようなフローチャートを作ってみよう。 ヒント



<実習 4-2：応用> いろいろな処理を使って、自由にフローチャートを作ってみよう。下の例は、ノベルゲームをイメージしたフローチャートです。

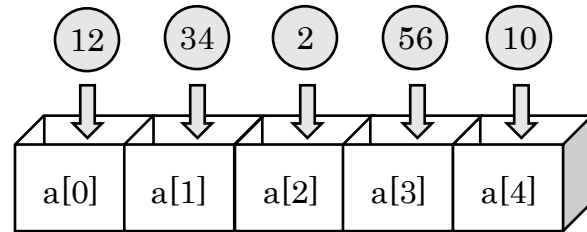


配列

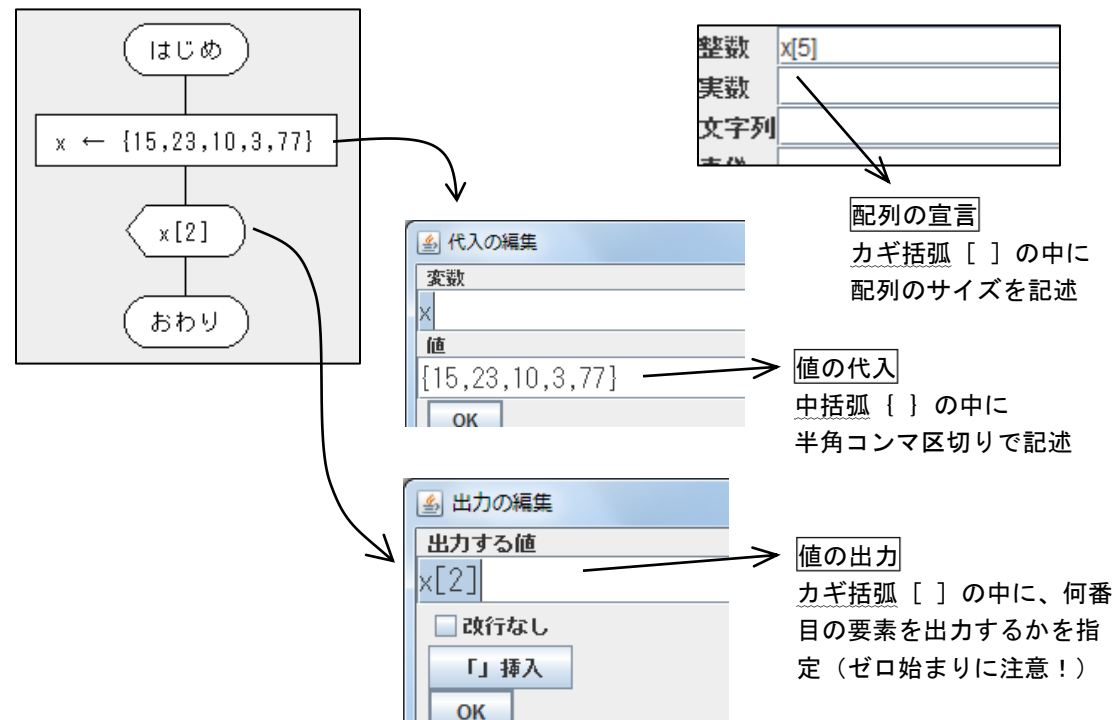
配列とは、変数の並びを一括して扱うことができるもので、並んだ箱のようなイメージです。一つの箱には一つのデータが入ります。

配列は、配列名 (a, score など) と、カギ括弧 [] で表す形式が多く、カギ括弧内に順番を表す番号 (0, 1, 2, ...) を指定します (言語によって異なる)。

下の図では、配列の名前は a、サイズは 5 で、番号は 0~4 となります。



<実習 5-1> 下のフローチャートを作成し、実行したら何が表示されるか考えよ。



<問> 結果表示は?

(答) 10

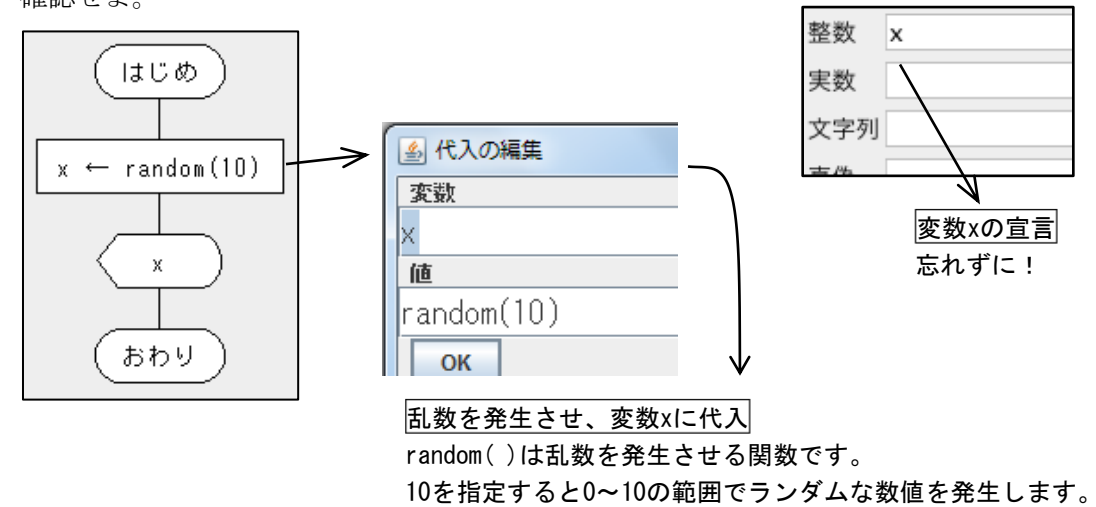
関数 (機能呼び出し)

関数とは、特定の機能や処理をまとめたもので、プログラミング言語側で用意されたもの (画面表示や数学関数など) と、自分で作成するものがあります。

関数は、関数名 (random, sleep など) と、中括弧 () で表す形式が多く、中括弧内に必要に応じて引数と呼ばれるデータを渡します (言語によって異なる)。関数の多くは、呼び出し元に返り値 (戻り値) と呼ばれる値を返します。

例: random(10) ... 0~10 までのランダムな整数を返す
sqrt(5) ... 5 の平方根を返す

<実習 5-2> 次のフローチャートを作成し、実行のたびに異なる数値が表示されることを確認せよ。



<応用実習 5-3> 実行するたびに結果が変わる、おみくじのフローチャートを作成してみよう。

ヒント: 実習 5-1 「配列」と実習 5-2 「乱数」を組み合わせる

注意: 文字列はカギ括弧「」でくくる。区切りのコンマは半角

{「大吉」,「吉」,「凶」}

発生させる乱数の範囲に注意! 3種類なら 0 番 1 番 2 番なので配列の宣言 m[] のカギ括弧内や random() の中括弧内に指定する数字は?