

# 配列(0)

配列基礎編

<http://java.cis.k.hosei.ac.jp/>

## 授業の前に自己点検

以下のことからを友達に説明できますか？

- 再帰呼び出しはどのように実行されますか
  - Pascalの三角形, Factorialの例で説明できますか
- 擬似コードの役割について説明できますか
- テストケース作成について説明できますか
  - 同値分割法
  - 境界値分析法

# 擬似コード(pseudocode)

- 擬似コードとは
  - 自然言語と通常のプログラミング言語を混ぜたようなコード (プログラム)
  - アルゴリズム辞典などでよく使われる
- 目的は
  - プログラムの処理や流れを簡単に記述するものである。
  - アイデアを整理したり他人に伝えるために用いる。
- 望ましい性質
  - どのプログラミング言語にも書き直しがしやすい

# 擬似コードの基本事項を適用してみる

## 「コンソールに入力された値の絶対値を表示する」

擬似コード

```
i nput = 入力された値  
pri nt(|i nput|)
```

Javaでの実現

```
package j1.lesson10;  
  
import java.io.*;  
public class Absolute {  
    public static void main(String[] args) throws IOException {  
        BufferedReader reader =  
            new BufferedReader(new InputStreamReader(System.in));  
        int input = Integer.parseInt(reader.readLine());  
        System.out.println(Math.abs(input));  
    }  
}
```

# 今回のテーマ

- 一次元配列
  - 同じ型のデータを一行に並べたもの

# 宣言と参照

配列変数の宣言と  
配列の初期化

```
int[] a = {3, 5, 7, 9, 11, 13};
```

参照はこのように

a[0] a[1] a[2] a[3] a[4] a[5]



これを実行すると

```
System.out.println(a[4]);
```

11



# 初期値を与えずに配列を生成する式

new 型の名前[要素数を表す式]

例えば

`int[] a = new int[10];`

`double[] b = new double[7];`

# 要素数の指定は変数でも式でもよい

```
BufferedReader reader =  
    new BufferedReader(new InputStreamReader(System.in));  
System.out.println("配列の大きさを入力:");  
int size = Integer.parseInt(reader.readLine());  
double[] c = new double[size];
```

配列の大きさは負になってはならない



# 配列の長さ

```
int[] a = {10, 20, 30};  
for (int i = 0; i < a.length; i++) {  
    System.out.println(a[i]);  
}
```

実行結果

10  
20  
30

↓  
擬似コードで書くと

```
a = {1, 2, 3}  
for i を 0 から (配列 a の長さ - 1) まで  
    print a[i]
```

最後の要素のインデックスは

# 一緒にやってみよう

- 今回の演習で使うテストドライバをいつものように指示通り正確にインストールする
  - テストドライバの導入に成功すると
    - プロジェクト「java20XX」の中の「test」というフォルダに「j1.lesson11basic.xml」という名前のファイルが作成される。
    - このファイルには今週使用するテスト一式が記述されている。
- j1.lesson11basic というパッケージを作成する
- 演習資料にあるShowArray, DynamicArray を作成し、一連のテストを手順通りに実行せよ。

# 課題

各自のペースで今回の課題をやってください。

# 課題1111のヒント

配列の宣言の仕方

```
double[] 配列変数 = new double[配列の長さ];
```

# 課題1112 最大の求め方

```
indexOfMaximum = 0          ←初代暫定チャンピオンの番号
maximum = data[0]          ←初代暫定チャンピオン

for i を 1 から dataSize - 1 まで    ←挑戦者に順に当たる
    if maximum < data[i] ←挑戦者が勝ったら(大きい方が勝ち)
        indexOfMaximum = i
        maximum = data[indexOfMaximum]
```

# 課題1112 最小の求め方

```
indexOfMinimum = 0      ←初代暫定チャンピオンの番号  
minimum = data[0]      ←初代暫定チャンピオン
```

```
for i を 1 から dataSize - 1 まで    ←挑戦者に順に当たる  
    if minimum > data[i] ←挑戦者が勝ったら(小さいほうが勝ち)  
        indexOfMinimum = i  
        minimum = data[indexOfMinimum]
```

# 課題1113 配列xの宣言

```
double[] x = new double[配列の長さ];
```

# 課題1113

## i番目とi+1番目の頂点間の距離

$$\sqrt{(x[i]-x[i+1])^2 + (y[i]-y[i+1])^2}$$

で計算する。平方は単純に掛け算で、平方根は

`Math.sqrt(...)`

を使う。

この式が通用するのは  $i = 0$  から  $i = \text{numberOfVertices} - 2$

(あるいは `x.length - 2` と言ってもよい。)

最後の点と最初の点を結ぶ式は別に必要。