

# オブジェクト指向プログラミング I 演習問題

2017 年 5 月 29 日

## 学習ポイント

- エラーの除去
- クラスの定義（フィールド・メソッド・コンストラクタ）
- インスタンスの生成・フィールドの参照・メソッドの呼び出し
- コマンドライン引数の使い方
- 配列の使い方

1. 配付のファイルをコンパイルするとエラーが出ます。

このエラーを除去して、プログラムを実行し、実行結果が以下になれば OK です。ただし、Main は変更できません。誤りの箇所および、修正方法を Scomb のアンケートで報告しなさい。

```
r1 長方形は、幅 = 10、高さ = 20、面積 = 200 です。
r2 長方形は、幅 = 12、高さ = 15、面積 = 180 です。
r3 長方形は、幅 = 30、高さ = 40、面積 = 1200 です。
r3 は r1 と面積が等しいか? = false
r3 長方形は、幅 = 25、高さ = 8、面積 = 200 です。
r3 は r1 と面積が等しいか? = false
```

2. x 座標と y 座標からなる点のクラス Point を定義し、以下に示すように座標を線形変換するメソッド linearTransfer を定義しなさい。コマンドライン引数で与えた x 座標と y 座標の値からなる点と線形変換後の点の座標を下記のように画面表示するプログラムを作成しなさい。

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 6 & 4 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

```
Input Point : (5.0, 10.0)
=> Point after linearTransfer : (70.0, 0.0)
```

※コマンドライン引数とは、前回の演習で用いた ShowArgs.java の実行時に与える引数です。

```
class ShowArgs{
    public static void main(String[] args){
        System.out.println("引数の個数 = " + args.length);
        for (int i = 0; i < args.length; i++){
            System.out.println(i+1 + "番目の引数は、" + args[i] + " です。");
        }
    }
}
```

たとえば、下記のような実行結果が得られます。

```
java -cp class ShowArgs 100 aa !!
引数の個数 = 3
1 番目の引数は、100 です。
2 番目の引数は、aa です。
3 番目の引数は、!! です。
```

コマンドライン引数の型は **String** ですが、この問題で必要な入力は、**double** 型です。  
そこで、型の変換は以下の **Double** クラスのメソッドを用いて行います。

```
public static parseDouble(String s)
```

パラメータ 文字列

戻り値 文字列引数で表される **double** 値

3. 幅 **2**、高さ **1** の長方形を1つ作成し、順次、幅、高さともに **2** 倍にした長方形を **15** 個作成し、**15** 個の長方形の幅、高さ、面積を画面表示するプログラムを作成しなさい。**15** 個の長方形は、配列に格納しなさい。

※配列の宣言

型 [ ] 変数名; 例えば `int[] array;` `String[] word;`

配列の生成式

`new` 型 [ 整数式 ]    例えば `int[] array = new int[10];`    `String[] word = new String[20];`

4. 「線分」は2つの端点で定義される。問題 2 で定義した「点」クラス **Point** を用いて「線分」を表す **LineSegment** クラスを定義し、線分の 2 つの端点を問題 2 で定義した線形変換で変換し、新たな線分を得るメソッド **linearTransfer** を定義しなさい。このクラスを用いて、1 つの線分を定義するのに必要な 4 つの値をコマンドライン引数で与えて、その線形変換後の線分を以下のように画面表示する **main** メソッドをもつクラス **Main** を定義しなさい。

```
Input LineSegment (5.0,10.0)-(10.0,20.0)
```

```
=> LineSegment after linearTransfer : (70.0, 0.0) -> (140.0,0.0)
```