

デジタル表現論・第4回

劉 雪峰 (リュウ シュウフォン)

2016年5月2日

本日の目標

Java プログラミングの基礎
出力の復習
メソッドの定義と使用

出力

`System.out.print()` と `System.out.println()` は文字列を端末に出力するためによく使用されるメソッドである。

変数 `s` を文字列とする。

- 1) `System.out.println(s)` は文字列 `s` の最後に自動的に改行 ("`\n`" = "`¥n`") を加える。
- 2) `System.out.print(s)` は改行なし、文字列 `s` を端末に出力する。
- 3) `System.out.println(s) = System.out.print(s + "\n")`

例：

print

```
s = " A "  
System.out.print(s);  
System.out.print(s);
```

出力:

A A

println

```
s = " A "  
System.out.println(s);  
System.out.println(s);
```

出力:

A
A

メソッドの使用

メソッドを試してみる

前回の授業の演習では、標準体重の計算をメソッドとしてまとめることができる。

```
import java.util.Scanner;
public class Code4_1{
    public static double Weight(double h){
        return h*h*22;
    }
    public static void main(String args[]){
        double height;
        Scanner scan = new Scanner(System.in);
        System.out.print("Input height (m) : ");
        height = scan.nextDouble();
        System.out.println("Standard weight:"+ Weight(height) + "kg");
    }
}
```

メソッドの定義

```
[修飾子] 戻り値のデータ型 メソッド名 (引数1, 引数2, ...) {  
    // メソッドの中身  
}
```

説明：

- 1) 修飾子は `public static` などを使用する。修飾子はそのメソッドがどこからアクセス可能なのかなどを示している。
- 2) 戻り値のデータ型は `int`, `double`, `String` など。戻り値がない場合、`void` を使用すること。
- 3) メソッド名は**アルファベット**または**アンダーライン**から始まる。数字で始まるのはいけない。

メソッドの呼び出し

メソッドの呼び出しは以下の形で行う。

```
// 戻り値のある場合  
変数 = メソッド名(引数1, 引数2, ...);  
// 戻り値のない場合  
メソッド名(引数1, 引数2, ...);
```

注意：メソッドの呼び出し元の変数とメソッド内の変数は互いに独立である。メソッド内の処理によって外部の変数を直接に修正することにはできない。

メソッドの使用例

```
public class Code4_2{
    public static void func1(){
        System.out.println("This is method 1.");
        System.out.println("No input and no output.");
    }
    public static int func2(int n){
        return 2*n;
    }
    public static void main (String[] args) {
        func1();
        int x=func2(10);
        System.out.println("The return value of func2 is:");
        System.out.println(x);
    } //main
} //Code4_2
```

プログラムの実行の流れに注意しなさい。

変数の有効範囲

メソッドの呼び出し元の変数とメソッド内の変数は互い独立である。
メソッド内の変数の有効範囲はメソッドの内部に限る。

以下の例の結果を確認しましょう。

```
public class Code4_3{
    public static void get_half(int n){
        n = n/2;
        System.out.println("In method get_half, n is "+n);
    }
    public static void main (String[] args) {
        int n=100;
        System.out.println("Befor calling, n is "+n);
        get_half(n);
        System.out.println("After calling, n is "+n);
    } //main
} //Code4_3
```

演習

- 1) ここまでのサンプルコード 4_1, 4_2, 4_3 を試してみて、メソッドの使用方法を理解してください。
- 2) 素数判定のコードにメソッドを導入する。

```
public class Report1{
    public static boolean prime_check (int n){
        boolean is_prime = true;

        ??

        return is_prime;
    } //prime_check

    public static void main (String[] args) {
        for(int n=2;n<100;n++){
            if( prime_check(n) ){
                System.out.println("Prime number:n="+n);
            }
        } //for loop with n
    } //main
} //Report1
```

メソッドの引数が配列の変数である場合

配列である変数は当該配列のアドレスを格納している。アドレスに対応する配列の要素の値は変更できる。

```
public class Code4_5{
    public static void set_array_value(int [] x){
        x[0]=1;
    } // set the first array element to 1
    public static void main (String[] args) {
        int [] x = new int [5];
        x[0]=10;
        System.out.println("Before:  " + x[0]);
        set_array_value(x);
        System.out.println("After:   " + x[0]);
    } //main
} //Code4_5
```

メソッドの引数が配列の変数である場合

配列である変数は当該配列のアドレスを格納している。アドレスに対応する配列の要素の値は変更できる。

```
public class Code4_5{
    public static void set_array_value(int [] x){
        x[0]=1;
    } // set the first array element to 1
    public static void main (String[] args) {
        int [] x = new int [5];
        x[0]=10;
        System.out.println("Before:  " + x[0]);
        set_array_value(x);
        System.out.println("After:  " + x[0]);
    } //main
} //Code4_5
```

出力: Before: 10 After: 1

レポート課題

以下のようなメソッドを定義しなさい。

- 1) メソッド名は `find_max_min` とする。
- 2) 引数は三つの整数である。
- 3) 三つの引数の中で、最大数と最小数を画面に出力する。

出力の例：

```
The maximum number is 30.  
The minimum number is 10.
```

ヒント

```
import java.util.Scanner; //入力用
public class Report2{
    public static void find_max_min(int x, int y, int z){
        double max_value, min_value;
        //最大値の計算
        //Step 1: xとyを比較して、大きい値をmax_valueに格納する。
        //Step 2: max_valueとzを比較して、大きい値をmax_valueに格納する。
        //Step 3: max_valueにはx,y,zの最大値が格納されているので、
        //max_valueを出力する。
    }
    public static void main (String[] args) {
        Scanner input = new Scanner(System.in); //入力用の変数
        int x,y,z;
        System.out.println("Please input x,y,z:");
        x = input.nextInt(); //整数を入力
        y = input.nextInt();
        z = input.nextInt();
        find_max_min(x,y,z);
    } //main
} //Report2
```

レポート課題 [2](オプション)

以下のようなメソッドを定義しなさい。

- 1) メソッド名は `sort_array` とする。
- 2) メソッドの引数は整数型の配列である。
- 3) 引数である配列の要素をソートして、小さいものから大きいものまでの順にする。
- 4) メソッドの戻り値はないようにする。
- 5) 配列の初期値は手動で設定するまたは端末から入力する。

出力の例：

```
Before sorting, the array is: 10 20 9 5 13;  
After sorting, the array is: 5 9 10 13 20.
```