

## 16 応用的なメソッド

```
1. import lib.Input;

public class P1 {

    public static void main(String[] args) {
        double s1 = Input.getDouble("縦");
        double s2 = Input.getDouble("横");
        double s3 = Input.getDouble("高さ");
        double w = Input.getDouble("重量");
        if(!isok(s1, s2, s3)){
            System.out.println("サイズオーバーで配送できない");
        }else{
            int charge = ryokin(s1,s2,s3,w);
            System.out.println("料金は" + charge + "円");
        }
    }
    // サイズチェック
    public static boolean isok(double a, double b, double c) {
        if(a+b+c <= 180) return true;
        return false;
    }
    // 料金計算
    public static int ryokin(double a, double b, double c, double w){
        int charge;
        double size = a+b+c;
        if(size<=90){
            if(w<=5){
                charge = 500;
            }else if(w<=10){
                charge = 1000;
            }else{
                charge = 1500;
            }
        }else{
            if(w<=5){
                charge = 1000;
            }else if(w<=10){
                charge = 2000;
            }else{
                charge = 3000;
            }
        }
        return charge;
    }
}
```

2.

```
import lib.Input;
public class P2 {
    public static void main(String[] args) {
        int n = Input.getInt();
        if(isJanken(n)){
            System.out.print(n + ":");
            switch(janken(n)){//値を返すメソッド呼び出しを書いてよい
                case 0:
                    System.out.println("引き分け");
                    break;
                case 1:
                    System.out.println("勝ち");
                    break;
                case 2:
                    System.out.println("負け");
                    break;
            }
        }else{
            System.out.println("入力した値が正しくない");
        }
    }
    // 1,2,3 の乱数を返す
    public static int rand3(){
        int r = (int)(Math.random()*3+1);
        return r;
    }
    // 1,2,3 のどれかなら OK
    public static boolean isJanken(int n){
        if(n==1||n==2||n==3){
            return true;
        }else{
            return false;
        }
    }
    // 勝敗表を 2 次元の配列にしたので要素を選択するだけで勝敗が分かる
    // n と m が配列要素番号になっているところがポイント
    public static int janken(int n){
        int[][] tbl = {
            {9,9,9,9 },
            {9,0,2,1 },
            {9,1,0,2 },
            {9,2,1,0 }
        };
        int m = rand3();
        System.out.print(m + " ");
        return tbl[m][n];
    }
}
```

3.

```
public class P3 {
    public static void main(String[] args) {
        int[] dt = {5,7,9,14,7,4,2};
        graph(dt);
    }
    // グラフを描きます
    public static void graph(int[] a){
        for(int n : a){
            dot(n);
        }
    }
    // ひとつの横棒グラフを描きます
    public static void dot(int n){
        for(int i=0; i<n; i++){
            System.out.print("#");
        }
        System.out.println(""); // 改行
    }
}
```

4.

```
public class P4 {
    public static void main(String[] args) {
        double[] dt = {15.1, 1.5, 17.2, 7.3, 21.0, 3.8, 6.8};
        double[] x = Tokei(dt);
        System.out.println("合計 = " + x[0]);
        System.out.println("平均 = " + x[1]);
        System.out.println("標準偏差 = " + x[2]);
    }
    public static double[] Tokei(double[] a){
        double[] ans = new double[3];
        ans[0] = total(a);
        ans[1] = mean(a);
        ans[2] = sd(a);
        return ans; // 配列 (の参照値) を返す
    }
    // 平均値を計算する
    public static double mean(double[] a){
        return total(a)/a.length;
    }
    // 合計を求める
    public static double total(double[] a){
        double total = 0;
        for(double d : a){
            total += d;
        }
        return total;
    }
    // 偏差の2乗の合計
    public static double vt(double[] a){
        double mean = mean(a);
        double vt = 0;
        for(double x : a){
            vt += Math.pow(x-mean,2);
        }
        return vt;
    }
    // 標準偏差
    public static double sd(double[] a){
        double vt = vt(a);
        return Math.sqrt(vt/a.length);
    }
}
```