14 多次元の配列

1. A. int[][] n = new int[][]{{1,2}, {3,4,5}};

   B. int[][] m = new int[10][];

2.
```
public class P2 {
     public static void main(String[] args) {
          int[][] n = {{1,2,3}, {4,5}, {6,7},{8}};
          for(int[] a : n){
               for(int m : a){
                    System.out.print(m+"¥t");
               }
               System.out.println("");//改行
          }
     }
}
```

3.
```
public class P3 {
     public static void main(String[] args) {
          int[][] n1 = { {10, 15, 22}, {8, 7, 12} };
          // コピー
          int[][] n2 = new int[n1.length][n1[0].length];
          for(int i=0; i<n2.length; i++){
               for(int j=0; j<n2[i].length; j++){
                    n2[i][j] = n1[i][j];
               }
          }
          // 表示
          for(int[] m : n2){
               for(int k : m){
                    System.out.print(k + "¥t");
               }
               System.out.println("");
          }
     }
}
```

4.

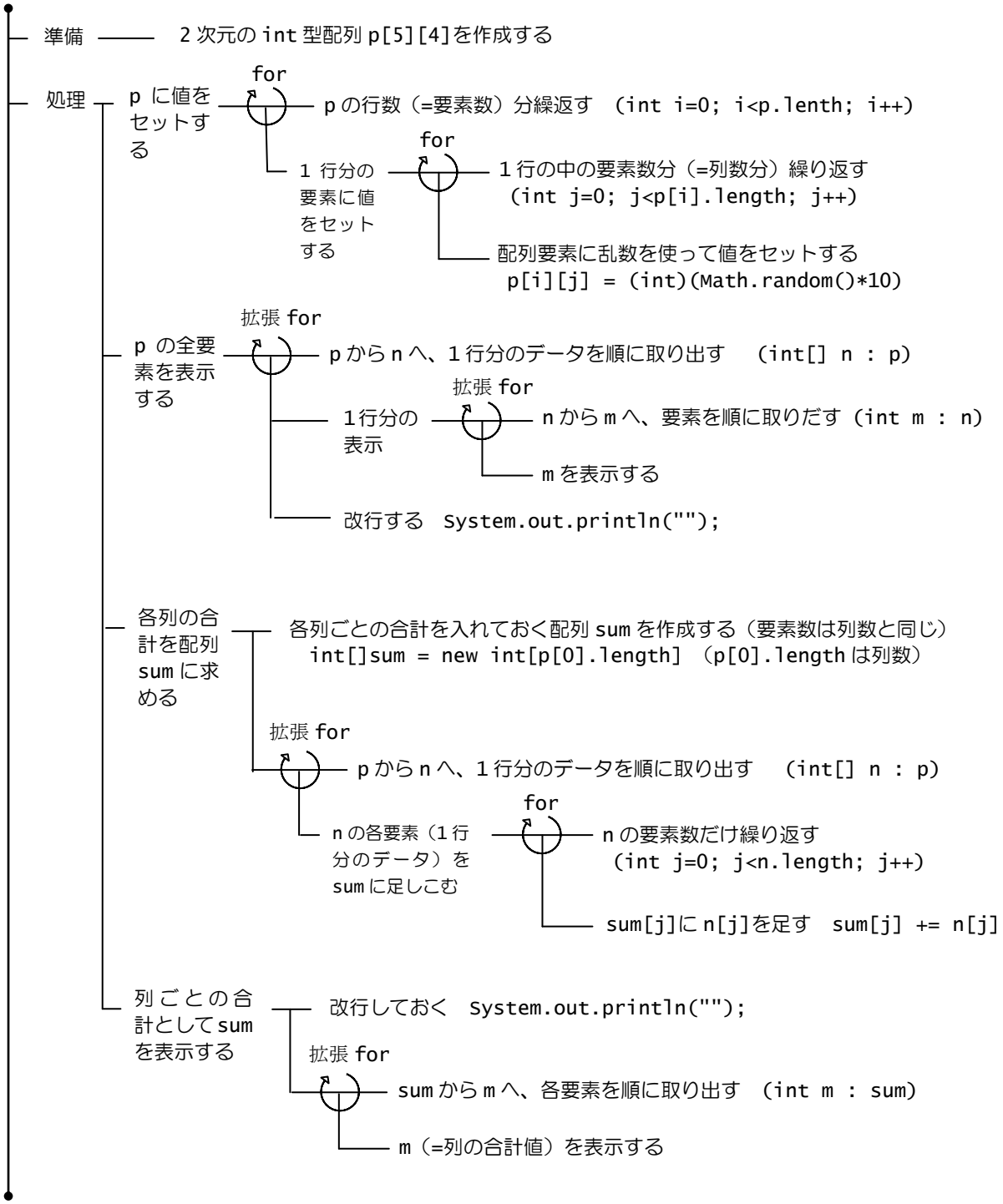
```
public class P4 {
     public static void main(String[] args) {
          int[][]  p = new int[5][4];
          // 乱数でデータ作成
          for(int i=0; i<p.length; i++){
               for(int j=0; j<p[i].length; j++){
                    p[i][j] = (int)(Math.random()*10);
               }
          }
          // データを表示
          for(int[] n : p){
               for(int m : n){
                    System.out.print(m + "¥t");
               }
               System.out.println("");
          }
          // 列の合計を取る
          int[] sum= new int[p[0].length];
          for(int[] n : p){
               for(int j=0; j<n.length; j++){
                    sum[j] += n[j];
               }
          }
          // 列の合計を表示
          System.out.println("");
          for(int m : sum){
               System.out.print(m + "¥t");
          }

     }
}
```

【解説】

やや複雑な処理なので、解答の SPD を次に示します。

プログラムの処理の意味を確認してください。


準備
2 次元の int 型配列 p[5][4]を作成する
処理
p に値をセットする
for
p の行数（=要素数）分繰返す （int i=0; i<p.lenth; i++)
1 行分の要素に値をセットする
for
１行の中の要素数分（=列数分）繰り返す
(int j=0; j<p[i].length; j++)
配列要素に乱数を使って値をセットする
p[i][j] = (int)(Math.random()*10)
p の全要素を表示する
拡張 for
p から n へ、１行分のデータを順に取り出す （int[] n : p)
1行分の表示
拡張 for
n から m へ、要素を順に取りだす (int m : n)
m を表示する
改行する System.out.println("");
各列の合計を配列 sum に求める
各列ごとの合計を入れておく配列 sum を作成する（要素数は列数と同じ）
int[]sum = new int[p[0].length] （p[0].length は列数）
拡張 for
p から n へ、１行分のデータを順に取り出す （int[] n : p)
n の各要素（1 行分のデータ）を sum に足しこむ
for
n の要素数だけ繰り返す
(int j=0; j<n.length; j++)
sum[j]に n[j]を足す sum[j] += n[j]
列ごとの合計として sum を表示する
改行しておく System.out.println("");
拡張 for
sum から m へ、各要素を順に取り出す (int m : sum)
m（=列の合計値）を表示する

5．A，D

【解説】B -- []は右側に書く、C -- int[][3]の部分は int[3][]である、E -- n と m は次元が違うので（つまり型が違う）ので参照値を代入できない、F -- new {} という書き方はない、G -- 初期化リストを与えているので int[2][]のように要素数を指定できない

6．B 【解説】m は 2 次元配列になる。m[0]と m[1]は同じ配列を共有する