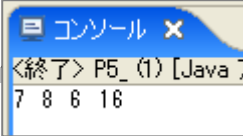


# 14章解答

1. 次は  $3 \times 4$  の配列の縦方向の合計を求めて表示するプログラムです。実行例のように表示するには、 と  に A~D のどれを入れたらよいか答えなさい。

```
public class P5_ {
    public static void main(String[] args) {
        int[][] n = {
            {2,1,3,6},
            {3,5,2,7},
            {2,2,1,3}
        };
        for(  ){
            int sum = 0;
            for(  ){
                sum += n[i][j];
            }
            System.out.print(sum + " ");
        }
    }
}
```



- A. `i=0; i<3; i++`  
B. `i=0; i<4; i++`  
C. `j=0; j<3; j++`  
D. `j=0; j<4; j++`

【解答欄】

@10点×2=20点

<input type="text" value="1"/>	D
<input type="text" value="2"/>	A

## 《解説》

横の並びを行といい、縦の並びを列という。プログラムでは、配列 `n` の要素を `n[i][j]` と書いているので、何行あるか（横の並びがいくつあるか）をカウントするのに `i` を使い、何列あるか（縦の並びがいくつあるか）をカウントするのに `j` を使うことになる。

列計を求めて順に表示する処理なので、外側の `for` 文は、一つの列（縦方向の計）の合計を求めるための反復である。列の数は、4列なので、(`for j=0; j<4; j++`)という繰り返しになる。

一方、内側の `for` 文は行をカウントするためのものであり、行は3行あるので、(`for(int i=0; i<3; i++`)という `for` 文になる。

# 14章解答

2. String の配列 s1 の要素が{ {"a", "bc", "def"}, {"xx","y" ,"zzz" } }であるとき、これをコピーした配列 s2 を作成しなさい。また、作成後、s2 のすべての要素を拡張 for 文で表示しなさい。

【解答欄】 20 点

```
public class P2 {
    public static void main(String[] args) {
        String[][] s1 = { {"a", "bc", "def"}, {"xx", "y", "zzz"} };
        String[][] s2 = new String[s1.length][s1[0].length];

        for(int i=0; i<s1.length; i++){
            for(int j=0; j<s1[i].length; j++){
                s2[i][j] = s1[i][j];
            }
        }

        for(String[] s : s2){
            for(String a : s){
                System.out.print(a + " ");
            }
            System.out.println("");
        }
    }
}
```

《解説》

配列のコピーでは、s2 = s1; としてはいけない。これはコピーではなく、S1とS2が同じ配列を共有することになるだけで、s1、s2のどちらを使っても同じ配列オブジェクトを操作することになる。

同じ値を持つ配列をもう一つ作るのがコピーである。したがって、新たに同じサイズの配列を new で生成し、そこへ要素をひとつずつコピーする操作が必要である。

# 14章解答

3. 次のA～Cの手順に従ってプログラムを作成しなさい。

A. 要素を3個持つ `int` の2次元配列 `n` を作成しなさい。ただし、各要素は4個の要素を持つ `int` の配列とします。

B. キーボードをタイプして、整数を入力し `n` のすべての要素に値を代入しなさい。

C. `n` の各要素を  $3 \times 4$  の表と見たとき、各列（縦方向の並び）の合計と各行（横方向の並び）の合計を例示のように表示しなさい。

【解答欄】 20 点

```
import lib.Input;
public class P3 {
    public static void main(String[] args) {
        int[][] n = new int[3][4]; // データを入力する配列
        ① for(int[] a : n){ // 横（行）1行文のデータ配列を取り出す
            for(int i=0; i<a.length; i++){
                a[i] = Input.getInt(); // データ入力
            }
        }
        int[] retu_kei = new int[n[0].length]; // 各列（縦）の合計を記録する
        配列
        ② for(int[] a : n){ // 横（行）1行文のデータ配列を取り出す
            int sum = 0; // 横（行）方向の合計
            for(int i=0; i<a.length; i++){
                sum += a[i]; // 横（行）方向の合計を計算
                retu_kei[i] += a[i]; // 縦（列）方向の合計を計算
                System.out.print(a[i]+"¥t"); // 横（行）1行のデータを表示
            }
            System.out.println(sum); // 横（行）方向の合計を表示
        }
        int total = 0; // 総合計
        ③ for(int m : retu_kei){
            total += m; // 総合計を計算
            System.out.print(m+"¥t"); // 縦（列）計を表示
        }
        System.out.println(total); // 総合計を表示
    }
}
```

《解説》

すべてを普通の `for` 文で処理するのではなく、下線を引いた箇所のように、横（行）1行分の配列を拡張 `for` 文で取り出して処理するようにすると、全体が簡単になります。

①の部分で配列 `n` に値を入力し、②の部分でデータとその横（行）計を表示します。③の部分では縦（列）合計と総合計を表示します。総合計は縦（列）計を合計することで求めています。

# 14章解答

3. 次のプログラムを書きなさい。

- A. char 型の配列{'a','b','c'}と{'d','e','f'}を要素にもつ配列 ch
- B. 10 個の要素を持つ double の配列の配列 (2 次元配列) x を作成する
- C. 2 個の要素を持つ int の配列の配列 (2 次元配列) n に、int 型の配列{1,2,3}と{4,5,6}を要素にもつ無名配列を代入する

【解答欄】 @5×3=15 点

A	char[][] ch = {'a', 'b', 'c'}, {'d', 'e', 'f'};
B	double[][] x = new double[10][];
C	int[][] n = new int[][]{{1, 2, 3}, {4, 5, 6}};

4. 次の配列変数の用法で正しいものをすべて答えなさい (複数ある)。

- A. double m=3; String[] n = new String[m]; m は double なので配列サイズ指定に使えない
- B. char c1[], c2='a'; OK
- C. byte[5] a = {-5,3,5,6,-2}; 要素を列記する時は、配列サイズは指定できない
- D. int[] n; n = {1,2,3,4,5}; 配列宣言の後で要素リストで配列を初期化できない
- E. double[] a; a=new double[] {1,2,3,4,5}; OK
- F. double a[] = new int[]{5,6,7}; 型が違う
- G. short []a = new short[2]; a = new {10,8}; new {10,8}; という書き方は文法エラー
- H. int []a, b[], []c; b[]は 2 次元配列になるが、[]c は文法エラー。
- I. int[] a = {1,2,3}; double[] b; b= a; 型が違うので配列変数同士の代入は不可
- J. char [][]n = new char[][5]; 最初の[]内の要素数は省略できない
- K. double [][]x = new double[2][3]; x[0]=null; OK

【解答欄】

15 点

B, E, K

5. 次のプログラムを実行したときの結果として正しいものはどれか答えなさい。

```
double[][] a = new double[2][3];
float[] b1 = {1.2f, 3.3f};
float[] b2 = null;
a[0] = (double[])b1;
a[1] = (double[])b2;
System.out.println(a[0][1]);
```

配列変数はキャストできない。

a[0] = (double[])b1;のような書き方はコンパイルエラーになる。

- A. 0
- B. 1.2
- C. 3.3
- D. null
- E. コンパイルエラーになる
- F. 実行時エラーになる

【解答欄】

10 点

E