

## 13 章 解答

1. 要素を 5 つ持つ **String** の配列 **s** を作成し、全要素にキーボードをタイプして入力した値を代入するプログラムを作成しなさい。最後に要素をすべて表示しなさい。

【解答欄】 @10

```
import lib.Input;
public class P1 {
    public static void main(String[] args) {
        String[] s = new String[5];    // 配列の作成
        for(int i=0; i<s.length; i++){ // 各要素に値を設定
            s[i] = Input.getString();
        }
        System.out.println(""); // 改行
        for(String str : s){ // 設定後の値を見る
            System.out.print(str + " ");
        }
    }
}
```

《解説》

①new に配列の作成方法、②要素数に **length** を使うこと、③**s[i]=〇〇 ;** のような要素への代入、など 配列の基本的な用法を理解しているかテストする問題です。

2. **double** の配列 **d** の要素が{10.5, 12.6, 8.7, 4.5}であるとき、これをコピーした配列 **x** を作成しなさい。また、作成後、**x** のすべての要素を拡張 **for** 文で表示しなさい。

【解答欄】 @10

```
public class P2{
    public static void main(String[] args) {
        double[] d = {10.5, 12.6, 8.7, 4.5};
        double[] x = new double[d.length];
        for(int i=0; i<x.length; i++){
            x[i] = d[i];
        }
        for(double a : x){
            System.out.print(a + "%t");
        }
    }
}
```

《解説》

配列 **d** を配列 **x** にコピーする際、**d = x ;** のように書いてしまいがちですが、これは同じ配列オブジェクトを、二つの変数で共有するだけで、コピーではありません。ひとつの配列オブジェクトに2つの名前を付けるようなものです。

コピーするには、解答のように、配列を新しく **new** で作成し、そこへ要素をひとつずつ代入してコピーします。

## 13 章 解答

3. `int` の配列の中に入力された値で割り切れる値があるかどうか検査するプログラムを作成します。`int` の配列として{102, 148, 203, 135, 77, 6}を要素に持つ `dt` を作成しておきます。

キーボードをタイプして `int` の値を 1 つ入力し、その値で割り切れる値が `dt` の要素の中にあるかどうか調べなさい。もしもあった場合は"あり"とコンソールに表示し、なければ"なし"とコンソールに表示しなさい。

【解答欄】 @15

```
import lib.Input;

public class P3 {
    public static void main(String[] args) {
        int[] dt = {102, 148, 203, 77, 6};
        int n = Input.getInt();
        String message = "なし";
        for(int i=0; i<dt.length; i++){
            if(dt[i]%n==0){
                message = "あり";
                break; // ループを脱出する
            }
        }
        System.out.println(message);
    }
}
```

《解説》

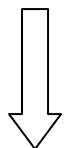
配列のサーチ（探索）問題です。発見した時、`for` 文の中から `break` で抜けます。ただし、その際、発見して抜けたことを記録しておくため、何かの変数に適当な値を入れて `break` します。ここでは、発見後に、"あり"を表示するので、`String` 型変数 `msg` に、最初は"なし"と入れておいて、`for` 文の中で発見したときだけ"あり"を代入するようにしています。

## 13 章 解答

4. `int` の配列 `a={1,2,3,4,5}` があるとき、これを `{5,1,2,3,4}` に並べ替えるプログラムを作成しなさい。ただし、配列は `a` 以外に作成してはいけません。

<説明>

- ・最初に `a[4]` を変数 `m` にいれて保存しておきます
- ・次に以下の操作を `for` 文で実行します。 <ヒント: `a[i+1]=a[i]` とします>

	<code>a[4] ← a[3]</code>	<code>a[4]</code> は保存してあるので変更してもよい
	<code>a[3] ← a[2]</code>	<code>a[3]</code> は <code>a[4]</code> に移したので変更してもよい
	<code>a[2] ← a[1]</code>	<code>a[2]</code> は <code>a[3]</code> に移したので変更してもよい
	<code>a[1] ← a[0]</code>	<code>a[1]</code> は <code>a[2]</code> に移したので変更してもよい

- ・最後に `a[0]←m` とします

【解答欄】 @15

```
public class P4 {
    public static void main(String[] args) {
        int[] a = {1, 2, 3, 4, 5};
        int m = a[4];
        for(int i=3; i>=0; i--){
            a[i+1] = a[i];
        }
        a[0] = m;
        for(int x : a){
            System.out.print( x + "¥t");
        }
    }
}
```

《解説》

配列要素の並べ替えはよくある問題です。

端の要素を別の変数に入れて保存しておき、その後、順にずらしていき、最後に保存しておいた値を適当な配列要素に戻せばいいわけです。

`for` 文のループ制御変数の使い方がポイントになります。

## 13 章 解答

5. 次のプログラムを実行した時の結果として正しいものはどれか答えなさい。

```
double[] a;  
for(double x : a){  
    System.out.println(x);  
}
```

- A. null が表示される
- B. 何も表示されない
- C. コンパイルエラー
- D. 実行時例外を発生する

【解答欄】

C

@10

《解説》

配列 `a` は宣言しただけで、`new` で実体を作成したわけではありません。つまり、初期化されていない配列です。したがって、その要素を `for( double x : a) { }` のように取り出そうとするコードは、初期化されていない配列を使おうとしているということでコンパイルエラーになります。

6. 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。

```
short[] m = {1,2,3,4,5};  
int[] n;  
n = m;  
for(int a : n){  
    System.out.println(a);  
}
```

- A. 12345
- B. 00000
- C. 何も表示されない
- D. コンパイルエラーになる
- E. 実行時エラーになる

【解答欄】

D

@10

《解説》

`m` は `short` 型の配列で、`n` は `int` 型の配列です。

配列変数は、型が違っていると互いに代入することはできませんので、これはコンパイルエラーになります。

## 13 章 解答

7. 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。

```
int[] a = {1,2,3,4,5}, b,c;  
c=b=a;  
c[0] = 0;  
for(int n : a){  
    System.out.print(n);  
}
```

- A. 12345
- B. 02345
- C. 何も表示されない
- D. コンパイルエラーになる
- E. 実行時エラーになる

【解答欄】

@10

B

《解説》

c=b=a; のような配列変数の代入は、コピーではなく、int[] a={1,2,3,4,5}という配列の実体に、bおよびcという別な名前を付けるだけの効果です。いわば、a,b,cの3つの変数で、同じ配列を共有したということです。  
従って c[0]=0 とすると、a[0]も b[0]も 0 です。

8. 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。ただし、左端の数字は行番号である。

```
1. double[] x = {2.1, 5.0, 3.3};  
2. int[] n = null;  
3. n = (int)x;  
4. for(int m : n){  
5.     System.out.print(m+ " ");  
6. }
```

- A. 2.1 5.0 3.3
- B. 2 5 3
- C. 2.0 5.0 3.0
- D. 3行目でコンパイルエラー
- E. 3行目で実行時例外
- F. 5行目でコンパイルエラー
- G. 5行目で実行時例外

【解答欄】

@10

D

《解説》

x は double 型の配列で、n は int 型の配列です。  
配列変数は、型が違っても互いに代入することはできませんが、キャストすることも不可能です。したがって 3 行目でコンパイルエラーになります。

## 13 章 解答

9. 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。@10

```
int[] val = {2,4,6}, n;  
n = val;  
int total=0;  
for(int i=0; i<n.length; i++){  
    val[i]++;  
    n[i]++;  
    total += n[i];  
}  
System.out.println("合計="+total);
```

- A. 12
- B. 15
- C. 18
- D. コンパイルエラーになる
- E. 実行時例外になる

【解答欄】

@10

C

《解説》

n=val;としているので、{2,4,6}をnとvalで共有しています。for文の中に、val[i]++とn[i]++があるので、同じ配列要素の値が一度に2増えることに注意してください。3回のループで2ずつ増えるので合計で12よりも6多い数になります。