

## プログラミング・ロジック 実習課題 解答例

---



この文書には、米国 Oracle Corporation 及び日本オラクル株式会社が権利を有する情報が含まれており、使用と開示に対して定められたライセンス契約に従って提供されるものです。また、これらは著作権法による保護も受けています。ソフトウェアのリバース・エンジニアリングは禁止されています。

この文書が合衆国政府の国防省関連機関に配布される場合は、次の制限付き権利が適用されます。

Use, duplication or disclosure by the Government is subject to restrictions for commercial computer software and shall be deemed to be Restricted Rights software under Federal law, as set forth in subparagraph (c) (1) (ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

当社の事前の書面による承諾なしに、いかなる形式あるいはいかなる方法でも、本書及び本書に付属する資料の全体又は一部を複製することを禁じます。いかなる複製も著作権法違反であり、民事または刑事、もしくは両者の制裁の対象となりえます。

この文書が合衆国政府の国防総省以外の機関に配布される場合は、その権利は、FAR 52.227-14「一般データにおける権利」(Alternate III を含む) (June 1987) で定める権利の制限を受けます。

この文書の内容は予告なく変更されることがあります。

米国 Oracle Corporation 及び日本オラクル株式会社は、本書及び本書に付属する資料についてその記載内容に誤りがない事及び特定目的に対する適合性に関するいっさいの保証を行うものではありません。また、本書を参考にアプリケーション・ソフトウェアを作成された場合であっても、そのアプリケーション・ソフトウェアに関して米国 Oracle Corporation 及び日本オラクル株式会社（その関連会社も含みます）は一切の責任を負いかねます。

ORACLE は、Oracle Corporation およびその関連会社の登録商標です。

本書で参照されているその他全ての製品やサービスの名称は、それぞれを表示する為だけに引用されており、それぞれ各社の商標である場合があります。

## 実習課題解答例

---



## 1 章【ステップ A-1】

1. プラットフォーム非依存
2. オブジェクト指向
3. ネットワーク指向
4. セキュリティ

## 1 章【ステップ A-2】

実行例のとおりに表示されているか確認してください。

## 1 章【ステップ A-3】

### HelloWorld.java

```
1. // HelloWorld.java
2.
3. class HelloWorld{
4.     public static void main(String[] args){
5.         System.out.println("Hello World");
6.     }
7. }
```

## 2 章【ステップ A-1】

### Variable.java

```
1. // Variable.java
2.
3. class Variable {
4.     public static void main(String[] args) {
5.
6.         int num;    // 変数の宣言
7.         num = 10;   // 変数に値を代入
8.
9.         //変数の値を表示
10.        System.out.println("num : " + num);
11.    }
12. }
```

## 2 章【ステップ A-2】

### ArraySqrt.java

```
1. // ArraySqrt.java
2.
3. class ArraySqrt {
4.     public static void main(String[] args) {
5.
6.         // 配列の宣言
7.         double[] array = new double[3];
8.
9.         // 配列に値を代入
10.        array[0] = 1.414;
11.        array[1] = 1.732;
12.        array[2] = 2.236;
13.
14.        // 配列の値を表示
15.        System.out.println("The Square of 2 : " + array[0] );
16.        System.out.println("The Square of 3 : " + array[1] );
17.        System.out.println("The Square of 5 : " + array[2] );
18.    }
19. }
```

## 2 章【ステップ A-3】

### DispString.java

```
1. // DispString.java
2.
3. class DispString {
4.     public static void main(String[] args) {
5.
6.         // 変数の宣言と初期値の代入
7.         String myString = "Hello Java World";
8.
9.         // 変数の値を表示
10.        System.out.println("myString : " + myString);
11.    }
12. }
```

## 2 章【ステップ A-4】

### CommandLineData.java

```
1. // CommandLineData.java
2.
3. class CommandLineData {
4.     public static void main(String[] args) {
5.
6.         // コマンドライン引数のひとつめを表示
7.         System.out.println("args[0] : " + args[0]);
8.     }
9. }
```

## 2 章【ステップ B-1】

### Expression.java

```
1. // Expression.java
2.
3. class Expression {
4.     public static void main(String[] args) {
5.
6.         // 変数の宣言と初期値の代入
7.         int abc = -100;
8.         long smallNum = -200L;
9.         float pi = 3.1415f;
10.        double largeNum = 6.022e23;
11.        boolean display = false;
12.        char alphabet = 'B';
13.        String str = "Welcome to Sun Microsystems";
14.        int[] i = new int[3];
15.        i[0] = 100; i[1] = 200; i[2] = 300;
16.
17.        // 変数の値を表示
18.        System.out.println("abc      : " + abc);
19.        System.out.println("smallNum : " + smallNum);
20.        System.out.println("pi       : " + pi);
21.        System.out.println("largeNum : " + largeNum);
22.        System.out.println("display  : " + display);
23.        System.out.println("alphabet : " + alphabet);
24.        System.out.println("str      : " + str);
25.        System.out.println("i[0]    : " + i[0]);
26.        System.out.println("i[1]    : " + i[1]);
27.        System.out.println("i[2]    : " + i[2]);
28.    }
29. }
```

## 2 章【ステップ B-2】

### ConnectedString.java

```
1. // ConnectedString.java
2.
3. class ConnectedString {
4.     public static void main(String[] args) {
5.
6.         // コマンドライン引数を連結させて表示
7.         System.out.println("Connected String : " + args[0] + "-" + args[1]);
8.     }
9. }
```

## 2 章【ステップ C-1】

### **DataType.java**

```
1. // DataType.java
2.
3. class DataType {
4.     public static void main(String[] args) {
5.
6.         // 変数の宣言と初期値の代入
7.         int number = 104;
8.         double decimal = 4.2;
9.         boolean truth = false;
10.        char letter = 'V';
11.        String message = "Good Morning";
12.
13.        // 変数の値を表示
14.        System.out.println("number : " + number );
15.        System.out.println("decimal : " + decimal );
16.        System.out.println("truth   : " + truth );
17.        System.out.println("letter  : " + letter );
18.        System.out.println("message : " + message );
19.    }
20. }
```

## 2 章【ステップ C-2】

### **PriceData.java**

```
1. // PriceData.java
2.
3. class PriceData {
4.     public static void main(String[] args) {
5.
6.         // 配列の宣言
7.         int priceData[] = new int[5];
8.
9.         // 配列に値を代入
10.        priceData[0] = 100;
11.        priceData[1] = 200;
12.        priceData[2] = 300;
13.        priceData[3] = 400;
14.        priceData[4] = 500;
15.
16.        // 配列の値を表示
17.        System.out.println("Price No1 : " + priceData[0] + " yen");
18.        System.out.println("Price No2 : " + priceData[1] + " yen");
19.        System.out.println("Price No3 : " + priceData[2] + " yen");
20.        System.out.println("Price No4 : " + priceData[3] + " yen");
21.        System.out.println("Price No5 : " + priceData[4] + " yen");
22.    }
23. }
```



## 2 章【ステップ C-3】

### DispUser.java

```
1. // DispUser.java
2.
3. class DispUser {
4.     public static void main(String[] args) {
5.
6.         // 変数の宣言と初期値の代入
7.         String name = "Duke";
8.         int age = 20;
9.
10.        // 変数の値を表示
11.        System.out.println("Your Name : " + name);
12.        System.out.println("Your Age  : " + age);
13.    }
14. }
```

## 2 章【ステップ C-4】

### UserData.java

```
1. // UserData.java
2.
3. class UserData {
4.     public static void main(String[] args) {
5.
6.         // コマンドライン引数の値を表示
7.         System.out.println("UserId      : " + args[0]);
8.         System.out.println("UserName    : " + args[1]);
9.         System.out.println("UserAddress : " + args[2]);
10.        System.out.println("UserPhone   : " + args[3]);
11.    }
12. }
```

## 2 章【ステップ C-5】

### FruitStore.java

```
1. // FruitStore.java
2.
3. class FruitStore {
4.     public static void main(String[] args) {
5.
6.         // 4店のみかんの在庫を格納する配列の宣言
7.         int tangerine[] = new int[4];
8.
9.         // 4店のりんごの在庫を格納する配列の宣言
10.        int apple[] = new int[4];
11.
12.        int total; // 在庫の合計値
13.
14.        // String型をint型へ変換して配列に代入
15.        tangerine[0] = Integer.parseInt(args[0]);
16.        tangerine[1] = Integer.parseInt(args[1]);
17.        tangerine[2] = Integer.parseInt(args[2]);
18.        tangerine[3] = Integer.parseInt(args[3]);
19.
20.        apple[0] = Integer.parseInt(args[4]);
21.        apple[1] = Integer.parseInt(args[5]);
22.        apple[2] = Integer.parseInt(args[6]);
23.        apple[3] = Integer.parseInt(args[7]);
24.
25.        // みかんの在庫の合計を計算
26.        total = tangerine[0] +
27.                tangerine[1] +
28.                tangerine[2] +
29.                tangerine[3];
30.
31.        // みかんの在庫の表示
32.        System.out.println("みかん");
33.        System.out.println("東京 : " + tangerine[0]);
34.        System.out.println("大阪 : " + tangerine[1]);
35.        System.out.println("福岡 : " + tangerine[2]);
36.        System.out.println("札幌 : " + tangerine[3]);
37.        System.out.println("合計 : " + total + "¥n");
38.
39.        // りんごの在庫の合計を計算
40.        total = apple[0] +
41.                apple[1] +
42.                apple[2] +
43.                apple[3];
44.
45.        // りんごの在庫の表示
46.        System.out.println("りんご");
47.        System.out.println("東京 : " + apple[0]);
48.        System.out.println("大阪 : " + apple[1]);
49.        System.out.println("福岡 : " + apple[2]);
50.        System.out.println("札幌 : " + apple[3]);
51.        System.out.println("合計 : " + total);
52.    }
53. }
```

## 2 章【ステップ C-6】

### DataInput.java

```
1.  // DataInput.java
2.
3.  class DataInput {
4.      public static void main(String[] args) {
5.          int i;
6.          double d;
7.          boolean b;
8.
9.          // String型からint型へ変換して代入
10.         i = Integer.parseInt(args[0]);
11.
12.         // String型からdouble型へ変換して代入
13.         d = Double.parseDouble(args[1]);
14.
15.         // String型からboolean型へ変換して代入
16.         b = Boolean.valueOf(args[2]).booleanValue();
17.
18.         System.out.println("int      : " + i);
19.         System.out.println("double   : " + d);
20.         System.out.println("boolean  : " + b);
21.     }
22. }
```

### 3 章【ステップ A-1】

**Divide.java**

```
1. // Divide.java
2.
3. class Divide {
4.     public static void main(String[] args) {
5.
6.         // コマンドライン引数を数値に変換
7.         int num1 = Integer.parseInt(args[0]);
8.         int num2 = Integer.parseInt(args[1]);
9.
10.        // 除算と剰余算を行う
11.        int divAns = num1 / num2;
12.        int remAns = num1 % num2;
13.
14.        // 結果の表示
15.        System.out.println(num1 + " / " + num2 + " = " + divAns);
16.        System.out.println("remainder : " + remAns);
17.    }
18. }
```

### 3 章【ステップ A-2】

- ① y は 5 、 x は 6 。
- ② y は 7 、 x は 7 。

### 3 章【ステップ A-3】

**IncreCheck.java**

```
1. // IncreCheck.java
2.
3. class IncreCheck {
4.     public static void main(String[] args) {
5.         int x, y;
6.         x = 5;
7.
8.         y = x++; // 後置
9.         System.out.println("y = x++ : x=" + x + " y=" + y);
10.
11.        y = ++x; // 前置
12.        System.out.println("y = ++x : x=" + x + " y=" + y);
13.    }
14. }
```

### 3 章【ステップ A-4】

- ① false
- ② true
- ③ false
- ④ false

### 3 章【ステップ A-5】

**RelationCheck.java**

```
1. // RelationCheck.java
2.
3. class RelationCheck {
4.     public static void main(String[] args) {
5.
6.         // 変数の宣言と初期値の代入
7.         int m_int = 10000, t_int = 20000;
8.         double m_double = 10000.0;
9.         float m_float = 10000.5f;
10.        char m_char = 'a', t_char = 'A';
11.
12.        // 関係演算の結果格納用変数
13.        boolean b;
14.
15.        // 関係演算と結果の表示
16.        b = ( m_int >= t_int );
17.        System.out.println("m_int >= t_int    : " + b);
18.
19.        b = ( m_int == m_double );
20.        System.out.println("m_int == m_double : " + b);
21.
22.        b = (m_int == m_float);
23.        System.out.println("m_int == m_float  : " + b);
24.
25.        b = (m_char == t_char);
26.        System.out.println("m_char == t_char   : " + b);
27.    }
28. }
```

### 3 章【ステップ A-6】

- ① 3.3333333
- ② 3
- ③ 2

### 3 章【ステップ A-7】

**CastCheck.java**

```
1. // CastCheck.java
2.
3. class CastCheck {
4.     public static void main(String[] args) {
5.
6.         int n = 10, result_one, result_two;
7.         float x;
8.
9.         // 演算
10.        x = (float)n / 3;
11.        result_one = (int)(1.6 + 1.8);
12.        result_two = (int)1.6 + (int)1.8;
13.
14.        // 演算結果を表示
15.        System.out.println("x          : " + x);
16.        System.out.println("result_one : " + result_one);
17.        System.out.println("result_two : " + result_two);
18.    }
19. }
```

### 3 章【ステップ B-1】

**Area.java**

```
1. // Area.java
2.
3. class Area {
4.     public static void main(String[] args) {
5.         double x = 0, y = 0, pi = 3.14;
6.
7.         // コマンドライン引数の値を数値に変換
8.         x = Double.parseDouble(args[0]);
9.         y = Double.parseDouble(args[1]);
10.
11.        // 計算結果を表示
12.        System.out.println("Input Width   : " + x);
13.        System.out.println("Input Height  : " + y);
14.
15.        System.out.println("Triangle area : " + x * y / 2);
16.        System.out.println("Square area   : " + x * y);
17.        System.out.println("Circle area   : " + x * x * pi);
18.    }
19. }
```

### 3 章【ステップ B-2】

**Casts.java**

```
1. // Casts.java
2.
3. class Casts {
4.     public static void main(String[] args) {
5.         byte b;
6.         int i = 5;
7.         double d;
8.
9.         d = i;
10.
11.        System.out.println("d : " + d);
12.
13.        d = 7.85;
14.
15.        // double 型の値を型変換して int 型変数に代入
16.        i = (int)d;
17.        System.out.println("i : " + i);
18.
19.        // int 型の値を型変換して byte 型変数に代入
20.        b = (byte)i;
21.        System.out.println("b : " + b);
22.    }
23. }
```

### 3 章【ステップ B-3】

**Weight.java**

```
1. // Weight.java
2.
3. class Weight {
4.     public static void main(String[] args) {
5.         double stdWeight; // 標準体重を格納
6.         double height;    // 身長を格納
7.
8.         // センチメートルをメートルに直す
9.         height = Double.parseDouble(args[0])/100;
10.
11.        // 標準体重 = 22 * 身長 * 身長
12.        stdWeight = 22 * height * height;
13.
14.        // 計算結果を表示
15.        System.out.println("あなたの標準体重 : " + stdWeight + " kg");
16.    }
17. }
```

### 3 章【ステップ C-1】

**TimeRequired.java**

```
1. // TimeRequired.java
2.
3. class TimeRequired {
4.     public static void main(String[] args) {
5.
6.         // コマンドライン引数を数値に変換
7.         int depTime = Integer.parseInt(args[0]);
8.         int depMin  = Integer.parseInt(args[1]);
9.         int arrTime = Integer.parseInt(args[2]);
10.        int arrMin  = Integer.parseInt(args[3]);
11.
12.        // 演算を行う
13.        int depTotal = depTime * 60 + depMin;
14.        int arrTotal = arrTime * 60 + arrMin;
15.
16.        int passage  = arrTotal - depTotal;
17.        int passTime = passage / 60;
18.        int passMin  = passage % 60;
19.
20.        // 結果の表示
21.        System.out.println("Time of Departure : " + args[0] + " : "+args[1]);
22.        System.out.println("Time of Arrival   : " + args[2] + " : "+args[3]);
23.        System.out.println("Time required    : " + passTime + "h "
24.                            + passMin + "min");
25.    }
26. }
```



### 3 章【ステップ C-2】

**AuthUser.java**

```
1. // AuthUser.java
2.
3. class AuthUser {
4.     public static void main(String[] args) {
5.
6.         // コマンドライン引数を数値に変換
7.         int idNumber      = Integer.parseInt(args[0]);
8.         int sectionNumber = Integer.parseInt(args[1]);
9.
10.        // IDナンバーと部門NOを値が正しいか調べる
11.        boolean resultCheck = (idNumber == 7777) && (sectionNumber == 3);
12.
13.        // 変数の値を標示
14.        System.out.println("ID Number      : " + idNumber);
15.        System.out.println("Section Number : " + sectionNumber);
16.        System.out.println("Authentication : " + resultCheck);
17.    }
18. }
```

### 3 章【ステップ C-3】

**ImposeTax.java**

```
1. // ImposeTax.java
2.
3. class ImposeTax {
4.     public static void main(String[] args) {
5.
6.         // コマンドライン引数を数値に変換
7.         int price = Integer.parseInt(args[0]);
8.
9.         // 消費税を算出
10.        int incTax = (int)(price * 1.08);
11.
12.        // 計算結果を表示
13.        System.out.println("Price : " + price );
14.        System.out.println("TaxIn : " + incTax );
15.    }
16. }
```

### 3 章【ステップ C-4】

**Change.java**

```
1.  // Change.java
2.
3.  class Change {
4.      public static void main(String[] args) {
5.          int price;    // 商品の値段を格納
6.          int change;   // おつりを格納
7.          int x,y;
8.
9.          price = Integer.parseInt(args[0]);
10.         change = 10000 - price;
11.
12.         System.out.println(price + " 円の商品の支払いを一万円札で行うと");
13.         System.out.println("おつりは、" + change + " 円になります ¥n");
14.
15.         //計算し、結果を表示
16.         x = change;
17.         y = x / 5000;
18.         System.out.println("五千円札 : " + y + "枚");
19.
20.         x = x - y * 5000;
21.         y = x / 1000;
22.         System.out.println("千円札   : " + y + "枚");
23.
24.         x = x - y * 1000;
25.         y = x / 500;
26.         System.out.println("五百円玉 : " + y + "枚");
27.
28.         x = x - y * 500;
29.         y = x / 100;
30.         System.out.println("百円玉   : " + y + "枚");
31.
32.         x = x - y * 100;
33.         y = x / 50;
34.         System.out.println("五十円玉 : " + y + "枚");
35.
36.         x = x - y * 50;
37.         y = x / 10;
38.         System.out.println("十円玉   : " + y + "枚");
39.
40.         x = x - y * 10;
41.         y = x / 5;
42.         System.out.println("五円玉   : " + y + "枚");
43.
44.         x = x - y * 5;
45.         y = x;
46.         System.out.println("一円玉   : " + y + "枚");
47.     }
48. }
```

## 4 章【ステップ A-1】

### EqualOrNot.java

```
1. // EqualOrNot.java
2.
3. class EqualOrNot {
4.     public static void main(String[] args) {
5.
6.         // コマンドライン引数の値を数値に変換
7.         int num1 = Integer.parseInt(args[0]);
8.         int num2 = Integer.parseInt(args[1]);
9.
10.        // 値の比較を行い、結果を表示する
11.        if( num1 == num2 ) {
12.            System.out.println("The numbers you input are same values");
13.        } else {
14.            System.out.println("The numbers you input are NOT same values");
15.        }
16.    }
17. }
```

## 4 章【ステップ A-2】

```
break;
```

## 4 章【ステップ A-3】

while 文では、繰り返し処理を実行する前に条件判定が行われるのに対して、do-while 文では、繰り返し処理を実行した後に条件判定が行われる。

## 4 章【ステップ A-4】

E

## 4 章【ステップ A-5】

### ForCheck.java

```
1.  // ForCheck.java
2.
3.  class ForCheck {
4.      public static void main(String[] args) {
5.
6.          // 0 ~ 5 未満の間、繰り返し
7.          for( int i = 0 ; i < 5 ;i++ ) {
8.              System.out.println("Java");
9.          }
10.     }
11. }
```

## 4 章【ステップ B-1】

### ScopeOfNum.java

```
1. // ScopeOfNum.java
2.
3. class ScopeOfNum {
4.     public static void main(String[] args) {
5.         int x = 0;
6.
7.         // コマンドライン引数がひとつの場合
8.         if( args.length == 1 ) {
9.             // コマンドライン引数の値を数値に変換
10.            x = Integer.parseInt(args[0]);
11.
12.            // 変換した値を表示
13.            System.out.println("Input Number : " + x);
14.
15.            // 変換した値が 0 ～ 10 未満の場合
16.            if( (x >= 0) && (x < 10) ){
17.                System.out.println(x + " is OK");
18.            } else {
19.                System.out.println(x + " is NG");
20.            }
21.            // コマンドライン引数がひとつではない場合
22.        } else {
23.            System.out.println("** Input error ** : Please input number");
24.        }
25.    }
26. }
```

## 4 章【ステップ B-2】

### Letter.java

```
1.  // Letter.java
2.
3.  class Letter {
4.      public static void main(String[] args) {
5.
6.          // コマンドライン引数がひとつの場合
7.          if( args.length == 1 ) {
8.              // 変換した値を表示
9.              System.out.println("Input a letter[A-E] : " + args[0]);
10.
11.             // 変換した値に対応した結果を表示
12.             switch(args[0]) {
13.                 case "A":
14.                     System.out.println("Excellent");
15.                     break;
16.                 case "B":
17.                     System.out.println("Very Good");
18.                     break;
19.                 case "C":
20.                     System.out.println("Good");
21.                     break;
22.                 case "D":
23.                     System.out.println("Fair");
24.                     break;
25.                 case "E":
26.                     System.out.println("Poor");
27.                     break;
28.                 default:
29.                     System.out.println("letter error:Unknown character " + args[0]);
30.             }
31.             // コマンドライン引数がひとつではない場合
32.             } else {
33.                 System.out.println("** Input error ** : Please input a letter");
34.             }
35.         }
36.     }
```

## 4 章【ステップ B-3】

### Star.java

```
1. // Star.java
2.
3. class Star {
4.     public static void main(String[] args) {
5.         int x = 0;
6.
7.         // コマンドライン引数がひとつ以上の場合
8.         if( args.length >= 1 ) {
9.             // コマンドライン引数の値を数値に変換
10.            x = Integer.parseInt(args[0]);
11.            // 変換した値を表示
12.            System.out.println("Input Number : " + x);
13.
14.            // 0 ~ 変換した値未満の間、繰り返し
15.            for( int i = 0; i < x; i++ ) {
16.                System.out.print("*");
17.            }
18.            System.out.println();
19.            // コマンドライン引数がひとつもない場合
20.        } else {
21.            System.out.println("** Input error ** : Please input number");
22.        }
23.    }
24. }
```

## 4 章【ステップ C-1】

### EvenOrOdd.java

```
1. // EvenOrOdd.java
2.
3. class EvenOrOdd {
4.     public static void main(String[] args) {
5.
6.         // コマンドライン引数の値を数値に変換
7.         int number = Integer.parseInt(args[0]);
8.         System.out.println("Input number : " + number);
9.
10.        int ans = number % 2;
11.
12.        // 計算結果が奇数
13.        if( ans != 0 ) {
14.            System.out.println("It's an odd number.");
15.            // 計算結果が偶数
16.        } else {
17.            System.out.println("It's an even number.");
18.        }
19.    }
20. }
```

## 4 章【ステップ C-2】

### BirthdaySeason.java

```
1. // BirthdaySeason.java
2.
3. class BirthdaySeason {
4.     public static void main(String[] args) {
5.
6.         // コマンドライン引数の値を数値に変換
7.         int birthdayMonth = Integer.parseInt(args[0]);
8.
9.         System.out.println("Input number          : " + birthdayMonth );
10.
11.        // 変換した数値が 1 ~ 12 ではなかった場合
12.        if( birthdayMonth < 1 || birthdayMonth > 12 ) {
13.            System.out.println("Please input again.");
14.            // 変換した数値が 1 ~ 12 の場合
15.        }else {
16.            String birthdaySeason = ""; // 季節
17.
18.            // 数値に対応した文字列を代入
19.            switch( birthdayMonth ) {
20.                case 3:
21.                case 4:
22.                case 5:
23.                    birthdaySeason = "Spring.";
24.                    break;
25.                case 6:
26.                case 7:
27.                case 8:
28.                    birthdaySeason = "Summer.";
29.                    break;
30.                case 9:
31.                case 10:
32.                case 11:
33.                    birthdaySeason = "Autumn.";
34.                    break;
35.                case 12:
36.                case 1:
37.                case 2:
38.                    birthdaySeason = "Winter.";
39.                    break;
40.            }
41.
42.            // 季節を表示
43.            System.out.println("Your birthday season : " + birthdaySeason );
44.        }
45.    }
46. }
```



## 4 章【ステップ C-3】

### CountDown.java

```
1. // CountDown.java
2.
3. class CountDown {
4.     public static void main(String[] args) {
5.         int count = Integer.parseInt(args[0]);
6.
7.         // 繰り返しの対象となる処理
8.         do {
9.             System.out.println("count : " + count );
10.            count--;
11.            // 0 以上の間、繰り返し
12.        } while (count >= 0) ;
13.
14.        System.out.println("finish!");
15.    }
16. }
```

## 4 章【ステップ C-4】

### LoopSum.java

```
1. // LoopSum.java
2.
3. class LoopSum {
4.     public static void main(String[] args) {
5.
6.         // コマンドライン引数の値を数値に変換
7.         int num = Integer.parseInt(args[0]);
8.
9.         System.out.println("Input number : " + num );
10.        System.out.println("-----");
11.
12.        // 数値が 0 以下の場合
13.        if( num <= 0 ) {
14.            System.out.println("Please input again.");
15.            // 数値が 0 以上の場合
16.        } else {
17.            int sum = 0;
18.
19.            // 1 ~ 数値以下の間、繰り返し
20.            for( int i=1; i<=num; i++ ) {
21.                sum = sum + i;
22.            }
23.
24.            // 計算結果を表示
25.            System.out.println("Total of addition : " + sum );
26.        }
27.    }
28. }
```

## 4 章【ステップ C-5】

### Star2.java

```
1.  // Star2.java
2.
3.  class Star2 {
4.      public static void main(String[] args) {
5.          int x = 0;
6.
7.          // コマンドライン引数がひとつ以上の場合
8.          if( args.length >= 1 ) {
9.              // コマンドライン引数の値を数値に変換
10.             x = Integer.parseInt(args[0]);
11.             // 変換した値を表示
12.             System.out.println("Input Number : " + x);
13.
14.             int i = 0;
15.
16.             // 変換した値未満の間、繰り返し
17.             while( i < x ) {
18.                 System.out.print("*");
19.                 i++;
20.             }
21.             System.out.println();
22.             // コマンドライン引数がひとつもない場合
23.         } else {
24.             System.out.println("** Input error ** : Please input number");
25.         }
26.     }
27. }
```

## 4 章【ステップ C-6】

## Star3.java

```
1. // Star3.java
2.
3. class Star3 {
4.     public static void main(String[] args) {
5.         int x = 0;
6.
7.         // コマンドライン引数がひとつ以上の場合
8.         if( args.length >= 1 ) {
9.             // コマンドライン引数の値を数値に変換
10.            x = Integer.parseInt(args[0]);
11.            // 変換した値を表示
12.            System.out.println("Input Number : " + x);
13.
14.            int i = 0;
15.
16.            // 無限ループ
17.            while( true ) {
18.                i++;
19.                // 条件を満たしたら繰り返しを抜ける
20.                if( i > x ) {
21.                    break;
22.                }
23.                System.out.print("*");
24.            }
25.            System.out.println();
26.            // コマンドライン引数がひとつもない場合
27.        } else {
28.            System.out.println("** Input error ** : Please input number");
29.        }
30.    }
31. }
```

## 4 章【ステップ C-7】

### Calc.java

```
1. // Calc.java
2.
3. class Calc {
4.     public static void main(String[] args) {
5.         long multiplier = 1; // べき乗の結果
6.
7.         System.out.println("*** 10 のべき乗を求めます ***");
8.
9.         // 1 ~ 10 の間、繰り返し
10.        for( int i = 1 ; i <= 10 ; i++ ) {
11.            // 計算結果を初期化
12.            multiplier = 1;
13.            System.out.print("10 の" + i + "乗は ");
14.            // べき乗の計算
15.            for( int j = 1 ; j <= i ; j++ ){
16.                multiplier *= 10;
17.            }
18.
19.            // 計算結果の表示
20.            System.out.println(multiplier + " です");
21.        }
22.    }
23. }
```

## 5 章【ステップ A-1】

### DispMessage.java

```
1. // DispMessage.java
2.
3. class DispMessage {
4.     public static void main(String[] args) {
5.
6.         // コマンドライン引数がひとつではない場合
7.         if( args.length != 1 ) {
8.             System.out.println("Please input again.");
9.         // コマンドライン引数がひとつの場合
10.        } else {
11.            // disp() メソッドの呼び出し
12.            disp(args[0]);
13.        }
14.    }
15.
16.    // disp() メソッドの定義
17.    static void disp(String message) {
18.
19.        // 引数の値を表示
20.        System.out.println("Message : " + message );
21.    }
22. }
```

## 5 章【ステップ A-2】

### CompareData.java

```
1.  // CompareData.java
2.
3.  class CompareData {
4.      public static void main(String[] args) {
5.
6.          // コマンドライン引数の値を int 型に変換
7.          int data1 = Integer.parseInt(args[0]);
8.          int data2 = Integer.parseInt(args[1]);
9.
10.         System.out.println("Input data 1 : " + data1 );
11.         System.out.println("Input data 2 : " + data2 );
12.
13.         // compData( ) メソッドの呼び出し
14.         int dataIndex = compData(data1, data2);
15.
16.         // 戻り値が 1 の場合
17.         if( dataIndex == 1 ) {
18.             System.out.println("First input data is larger.");
19.         // 戻り値が 2 の場合
20.         } else if( dataIndex == 2 ) {
21.             System.out.println("Second input data is larger.");
22.         // 戻り値が 0 の場合
23.         } else {
24.             System.out.println("Two input data is the same number.");
25.         }
26.     }
27.
28.     // compData( ) メソッドの定義
29.     static int compData(int num1, int num2) {
30.         int index = 0;
31.
32.         // num1 の値が大きい場合、1
33.         if( num1 > num2 ) {
34.             index = 1;
35.         // num2 の値が大きい場合、2
36.         } else if( num1 < num2 ) {
37.             index = 2;
38.         // それ以外(num1 と num2 が同じ値) の場合、0
39.         } else {
40.             index = 0;
41.         }
42.
43.         // 代入した値を返す
44.         return index;
45.     }
46. }
```

## 5 章【ステップ B-1】

## Commission.java

```
1.  // Commission.java
2.
3.  class Commission {
4.      public static void main(String[] args) {
5.
6.          // コマンドライン引数の値を int 型に変換
7.          int payment = Integer.parseInt(args[0]);
8.
9.          System.out.println("Your Payment      : " + payment );
10.
11.         // 変換した値が 1000 以上の場合
12.         if( payment >= 1000 ) {
13.
14.             // calcComm( ) メソッドの呼び出し
15.             double result = calcComm(payment);
16.
17.             // 計算結果の表示
18.             System.out.println("Commission included : " + result );
19.         // 条件を満たさない場合
20.         } else {
21.             System.out.println("Please input 1000 or more.");
22.         }
23.     }
24.
25.     // calcComm( ) メソッドの定義
26.     static double calcComm(int data) {
27.         double ans = 0;
28.
29.         // 引数の値が 2000 未満の場合
30.         if( data < 2000 ) {
31.             ans = data * 1.1;
32.         // 引数の値が 2000 未満ではなく、5000 未満の場合
33.         } else if( data < 5000 ) {
34.             ans = data * 1.05;
35.         // どの条件も満たさない場合
36.         } else {
37.             ans = data * 1.03;
38.         }
39.
40.         // 計算結果を返す
41.         return ans;
42.     }
43. }
```

## 5 章【ステップ C-1】

### ArgsCheck.java

```
1.  // ArgsCheck.java
2.
3.  class ArgsCheck {
4.      public static void main(String[] args) {
5.
6.          // コマンドライン引数がひとつではない場合
7.          if( args.length != 1 ) {
8.              System.out.println("Wrong Arguments");
9.          // コマンドライン引数がひとつの場合
10.         } else {
11.             // コマンドライン引数の値を int 型に変換
12.             int number = Integer.parseInt(args[0]);
13.             // check( ) メソッドの呼び出し
14.             boolean b = check(number);
15.
16.             // check( ) メソッドの戻り値を判断して結果を表示
17.             if( b ) {
18.                 System.out.println(number + " is OK!");
19.             } else {
20.                 System.out.println(number + " is NG!");
21.             }
22.         }
23.     }
24.
25.     // check( ) メソッドの定義
26.     static boolean check(int x) {
27.         boolean b;
28.
29.         // 引数の値が 100 以上 60000 以下の場合は、true
30.         if( (x >= 100) && (x <= 60000) ) {
31.             b = true;
32.             // 条件を満たさない場合は、false
33.         } else {
34.             b = false;
35.         }
36.
37.         // 代入した値を返す
38.         return b;
39.     }
40. }
```



## 5 章【ステップ C-2】

## Rectangle.java

```
1.  // Rectangle.java
2.
3.  class Rectangle {
4.      public static void main(String[] args) {
5.          int h = 0, w = 0;  // 縦、横
6.
7.          // コマンドライン引数がふたつの場合
8.          if( args.length == 2 ) {
9.              // コマンドライン引数の値を int 型に変換
10.             h = Integer.parseInt(args[0]);
11.             w = Integer.parseInt(args[1]);
12.
13.             // drawRect( ) メソッドの呼び出し
14.             drawRect(h, w);
15.             // コマンドライン引数がふたつではない場合
16.         } else {
17.             System.out.println("Wrong parameters");
18.         }
19.     }
20.
21.     // drawRect( ) メソッドの定義
22.     static void drawRect(int h, int w) {
23.         // 縦の値分、繰り返し
24.         for( int i = h; i > 0; i-- ) {
25.             // 横の値分、繰り返し
26.             for( int j = w; j > 0; j-- ) {
27.                 System.out.print("#");
28.             }
29.             System.out.print("\n");
30.         }
31.     }
32. }
```

## 5 章【ステップ C-3】

### Saving.java

```
1.  // Saving.java
2.
3.  class Saving {
4.      public static void main(String[] args) {
5.          double interest = 0, saving = 0; // 金利率、利息計算後
6.          long principal = 0;             // 元金
7.          int term = 0;                   // 利息期間
8.
9.          // コマンドライン引数がみつではない場合
10.         if( args.length != 3 ) {
11.             System.out.println("Wrong parameters");
12.             System.out.println
13.                 ("Please input parameters:principal interest term");
14.         // コマンドライン引数がみつの場合
15.         } else {
16.             // コマンドライン引数の値を 各データ型に変換
17.             principal = Long.parseLong(args[0]);
18.             interest  = Double.parseDouble(args[1]);
19.             term      = Integer.parseInt(args[2]);
20.
21.             // calculate( ) メソッドの呼び出し
22.             saving = calculate(principal, interest + 1, term);
23.
24.             // 各値、及び計算結果の表示
25.             System.out.println("Principal      : " + principal);
26.             System.out.println("Interest rate : " + interest);
27.             System.out.println("Term          : " + term);
28.             System.out.println("Now Saving    : " + (long)saving);
29.         }
30.     }
31.
32.     // calculate 利息計算
33.     static double calculate(long p, double i, int t) {
34.
35.         // 利息期間分、繰り返し
36.         for( int j = 1; j <= t; j++ ) {
37.             // 元金に金利を掛けた値を、元金に代入
38.             p *= i;
39.         }
40.
41.         // 元金を返す
42.         return p;
43.     }
44. }
```

## 5 章【ステップ C-4】

## Random.java

```
1.  // Random.java
2.
3.  class Random {
4.      public static void main(String[] args) {
5.          double num = 0;
6.
7.          // コマンドライン引数がひとつの場合
8.          if( args.length == 1 ) {
9.
10.             // コマンドライン引数の値を double 型に変換
11.             num = Double.parseDouble(args[0]);
12.             System.out.println("Input number      : " + num);
13.
14.             // ランダムな値を取得し、乗算
15.             num *= Math.random();
16.             System.out.println("Calculated number : " + num);
17.         } else {
18.             System.out.println("Wrong Parameters");
19.         }
20.     }
21. }
```

## 5 章【ステップ C-5】

### Exchange.java

```
1.  // Exchange.java
2.
3.  class Exchange {
4.      public static void main(String[] args) {
5.
6.          // コマンドライン引数がひとつの場合
7.          if( args.length == 1 ) {
8.
9.              // コマンドライン引数の値を double 型に変換
10.             double yen = Double.parseDouble(args[0]);
11.
12.             System.out.println(yen + "円を各国通貨に両替すると次のようになります¥n");
13.
14.             // 各メソッドの呼び出し
15.             System.out.println(Math.floor(dollar(yen) + 0.5) + "ドル");
16.             System.out.println(Math.floor(won(yen) + 0.5) + "ウォン");
17.             System.out.println(Math.floor(euro(yen) + 0.5) + "ユーロ");
18.             System.out.println(Math.floor(rupiah(yen) + 0.5) + "ルピア");
19.         } else {
20.             System.out.println("1つの数字を入力してください");
21.         }
22.     }
23.
24.     // 円を米ドルに変換するメソッドの宣言
25.     static double dollar(double yen) {
26.         return yen / 123;
27.     }
28.
29.     // 円を韓国ウォンに変換するメソッドの宣言
30.     static double won(double yen) {
31.         return yen * 10;
32.     }
33.
34.     // 円をユーロに変換するメソッドの宣言
35.     static double euro(double yen) {
36.         return yen / 121;
37.     }
38.
39.     // 円をインドネシア・ルピアに変換するメソッドの宣言
40.     static double rupiah(double yen) {
41.         return yen * 100;
42.     }
43. }
```

## 総合演習【ステップ A-1】

## Message.java

```
1.  // Message.java
2.
3.  class Message {
4.      public static void main(String[] args) {
5.
6.          // コマンドライン引数の数だけ繰り返し
7.          for( String str : args ) {
8.              // コマンドライン引数の値を順番に表示
9.              System.out.println(str);
10.         }
11.     }
12. }
```

## 総合演習【ステップ A-2】

## Average.java

```
1.  // Average.java
2.
3.  class Average {
4.      public static void main(String[] args) {
5.
6.          // コマンドライン引数がみつではない場合
7.          if( args.length != 3 ) {
8.              System.out.println("3つの整数を入力してください");
9.              System.exit(-1);
10.         }
11.
12.         // コマンドライン引数の値を int 型に変換
13.         int math = Integer.parseInt(args[0]);
14.         int jap = Integer.parseInt(args[1]);
15.         int eng = Integer.parseInt(args[2]);
16.
17.         double average; // 平均点
18.
19.         // 3教科の平均点を演算して代入
20.         average = ( ( math + jap + eng ) / 3.0 );
21.
22.         // 各値を表示
23.         System.out.println("数学・国語・英語の点数を計算します。");
24.         System.out.println("数学の点数      : " + math);
25.         System.out.println("国語の点数      : " + jap);
26.         System.out.println("英語の点数      : " + eng);
27.         System.out.println("3教科の平均点 : " + average);
28.     }
29. }
```

## 総合演習【ステップ A-3】

## Greeting.java

```
1.  // Greeting.java
2.
3.  class Greeting {
4.      public static void main(String[] args) {
5.
6.          // コマンドライン引数のチェック
7.          if ( args.length != 1) {
8.              System.out.println("0 から 23 までの時刻を 1 つ入力して下さい。");
9.              System.exit(-1);
10.         }
11.
12.         // コマンドラインから渡されたデータを int 型に変換
13.         int hour = Integer.parseInt(args[0]);
14.
15.         // 変換した数値が 0 以上 24 以下の場合、時刻に合わせたメッセージ出力
16.         if ( hour >= 0 && hour < 24 ) {
17.             switch (hour) {
18.                 // 数値が 6 から 11 の場合
19.                 case 6:
20.                 case 7:
21.                 case 8:
22.                 case 9:
23.                 case 10:
24.                 case 11:
25.                     System.out.println("おはようございます。");
26.                     break;
27.                 // 数値が 12 から 18 の場合
28.                 case 12:
29.                 case 13:
30.                 case 14:
31.                 case 15:
32.                 case 16:
33.                 case 17:
34.                 case 18:
35.                     System.out.println("こんにちは。");
36.                     break;
37.                 // 数値が 0 以上 24 以下で上記以外の場合
38.                 default:
39.                     System.out.println("こんばんは。");
40.             }
41.         } else {
42.             // 数値が 0 以上 24 以下でない場合
43.             System.out.println("入力された時刻が間違っています。");
44.         }
45.     }
46. }
```

## 総合演習【ステップ B-1】

## SunShop.java

```
1.  // SunShop.java
2.
3.  class SunShop {
4.      public static void main(String[] args) {
5.          int index = 0;  // 商品番号
6.          String name;
7.
8.          // コマンドライン引数がひとつの場合
9.          if( args.length == 1 ) {
10.             // コマンドライン引数の値を int 型に変換
11.             index = Integer.parseInt(args[0]);
12.          } else {
13.             System.out.println("ご注文希望の商品番号を入力して下さい。");
14.             System.exit(-1);
15.          }
16.
17.          // 注文希望商品の定義
18.          switch( index ){
19.              case 0:
20.                  name = "Duke 時計";
21.                  break;
22.              case 1:
23.                  name = "Java One T シャツ";
24.                  break;
25.              case 2:
26.                  name = "Duke トレーナー";
27.                  break;
28.              case 3:
29.                  name = "Duke ペン";
30.                  break;
31.              case 4:
32.                  name = "Java キャップ";
33.                  break;
34.              case 5:
35.                  name = "Sun ドライバーキット";
36.                  break;
37.              case 6:
38.                  name = "Sun バッグ";
39.                  break;
40.              case 7:
41.                  name = "Duke 人形";
42.                  break;
43.              default:
44.                  name = "0 から 7 までの商品番号指定が必要";
45.                  break;
46.          }
47.
48.          // 希望商品の標準出力
49.          System.out.println("ご希望の商品は : " + name + "です。");
50.      }
51.  }
```



## 総合演習【ステップ B-2】

## SalesTax.java

```
1.  // SalesTax.java
2.
3.  class SalesTax {
4.      public static void main(String[] args) {
5.          // 金額、年、月、税額用の変数宣言
6.          int price = 0, year = 0, month = 0, tax = 0;
7.          // 税率用の変数宣言
8.          double rate = 0.0;
9.
10.         // コマンドライン引数チェック
11.         if( args.length == 3 ) {
12.             // コマンドラインから入力された各データを数値に変換
13.             price = Integer.parseInt(args[0]);
14.             year  = Integer.parseInt(args[1]);
15.             month = Integer.parseInt(args[2]);
16.
17.             // 数値に変換した各データを表示
18.             System.out.println("Input Price : " + price);
19.             System.out.println("Input Year  : " + year);
20.             System.out.println("Input Month : " + month);
21.             System.out.println();
22.
23.             // 2014 年の 4 月以降 12 月までか、または 2014 年より先かどうか
24.             if( ((year == 2014) && (month >= 4) && (month <= 12))
25.                 || (year > 2014) ) {
26.                 // 税率を 0.08 とする
27.                 rate = 0.08;
28.                 // 上記以外
29.             } else {
30.                 // 税率を 0.05 とする
31.                 rate = 0.05;
32.             }
33.
34.             // 税額を計算する
35.             tax = (int)(price * rate);
36.             System.out.println("Sales Tax   : " + tax);
37.         } else {
38.             System.out.println("Please input Price, Year, Month");
39.         }
40.     }
41. }
```

## 総合演習【ステップ B-3】

## SearchData.java

```
1.  // SearchData.java
2.
3.  class SearchData {
4.      public static void main(String[] args) {
5.
6.          // コマンドラインから渡されたデータを数値に変換
7.          int inputData = Integer.parseInt(args[0]);
8.
9.          System.out.println("Input data : " + inputData );
10.
11.         // 予め指定された数値を設定する配列を宣言
12.         int[] dataBase = new int[5];
13.         dataBase[0] = 1111;
14.         dataBase[1] = 3333;
15.         dataBase[2] = 5555;
16.         dataBase[3] = 7777;
17.         dataBase[4] = 9999;
18.
19.         // フラグの宣言
20.         int flag = 0;
21.         int i;
22.
23.         // 用意した配列の大きさ分繰り返し処理
24.         for( i = 0; i < dataBase.length; i++ ) {
25.             // コマンドラインから渡されたデータを
26.             // 配列 database の各要素と比較
27.             // 値が同じ場合は flag を 1 にし、ループから抜ける
28.             if (dataBase[i] == inputData) {
29.                 flag = 1;
30.                 break;
31.             }
32.         }
33.
34.         if( flag == 1 ) {
35.             System.out.println("Data was found ! index: " + i );
36.         } else {
37.             System.out.println("Data was not found.");
38.         }
39.     }
40. }
```

## 総合演習【ステップ B-4】

## Triangle.java

```
1.  // Triangle.java
2.
3.  class Triangle {
4.      public static void main(String[] args) {
5.          if( args.length != 1) {
6.              System.out.println("Please input one number.");
7.              System.exit(-1);
8.          } else {
9.              // コマンドラインから入力された値を数値に変換
10.             int len = Integer.parseInt(args[0]);
11.             System.out.println("Input number : " + len);
12.
13.             // 変換された値が 1 より小さい場合
14.             if( len < 1 ) {
15.                 System.out.println("Please input again.");
16.             } else {
17.                 // 変換した数値の数分を底辺とした三角形を描く
18.                 for( int i = 0; i < len ; i++ ) {
19.                     for( int j = 0; j <= i; j++ ) {
20.                         System.out.print("#");
21.                     }
22.                     System.out.println();
23.                 }
24.             }
25.         }
26.     }
27. }
```

## 総合演習【ステップ C-1】

## Factorial.java

```
1.  // Factorial.java
2.
3.  class Factorial {
4.      public static void main(String[] args) {
5.
6.          // コマンドライン引数がひとつではない場合
7.          if( args.length != 1 ){
8.              System.out.println("データを入力して下さい!");
9.              System.exit(-1);
10.         }
11.
12.         // コマンドライン引数を int 型に変換
13.         int n = Integer.parseInt(args[0]);
14.
15.         // 変換した値が 0 以下の場合
16.         if( n <= 0 ){
17.             System.out.println(" 0 より大きい値を入力して下さい。");
18.             System.exit(-1);
19.         } else {
20.             // factorial( ) メソッドの呼び出し
21.             int e = factorial(n);
22.             System.out.println( n + " の階乗を表示します。: " + e );
23.         }
24.     }
25.
26.     // factorial( ) メソッドの定義
27.     static int factorial(int n) {
28.         int fact = 1;
29.
30.         // 階乗の演算を行う
31.         for( int i = n; i > 0; i-- ) {
32.             fact *= i;
33.         }
34.
35.         // 計算結果を返す
36.         return fact;
37.     }
38. }
```

## 総合演習【ステップ C-2】

## Euclid.java

```
1.  // Euclid.java
2.
3.  class Euclid {
4.      public static void main(String[] args) {
5.
6.          // コマンドライン引数がふたつではない場合
7.          if( args.length != 2 ) {
8.              System.out.println("2つの整数を入力してください");
9.              System.exit(-1);
10.         }
11.
12.         // コマンドライン引数の値を int 型に変換
13.         int a = Integer.parseInt(args[0]);
14.         int b = Integer.parseInt(args[1]);
15.
16.         int c;
17.
18.         // a の値が b より小さい場合、値を入れ替える
19.         if( a < b ) {
20.             c = a;
21.             a = b;
22.             b = c;
23.         }
24.
25.         // 最大公約数を求める
26.         do {
27.             c = a % b;
28.             a = b;
29.             b = c;
30.         } while( c != 0 );
31.
32.         System.out.println("2つの整数の最大公約数 = " + a);
33.     }
34. }
```

## 総合演習【ステップ C-3】

## PrimeNumber.java

```
1.  // PrimeNumber.java
2.
3.  class PrimeNumber {
4.      public static void main(String[] args) {
5.          int limit = 0;
6.
7.          // コマンドラインからのデータをチェックする
8.          if( args.length == 1 ) {
9.              limit = Integer.parseInt(args[0]);
10.
11.              if( limit < 2 ) {
12.                  System.out.println(" 2 より大きい数を入力してください。");
13.                  System.exit(-1);
14.              }
15.
16.              // 入力されたデータを標準出力する
17.              System.out.println( limit + " 以下の素数を求めます。");
18.
19.              // boolean 型の配列を宣言する
20.              boolean[] en = new boolean[limit + 1];
21.
22.              int i,j;
23.
24.              // 素数か否かの判定をする
25.              for( i = 0; i < en.length; i++ ) {
26.                  en[i] = true;
27.              }
28.
29.              for( i = 2; i <= limit; i++ ) {
30.                  if(en[i]) {
31.                      // 素数を標準出力する
32.                      System.out.println("素数 : " + i );
33.                      for( j = i + i; j <= limit; j += i ) {
34.                          en[j] = false;
35.                      }
36.                  }
37.              }
38.          }
39.      }
40.  }
```

## 総合演習【ステップ C-4】

## SunJapanShop.java

```
1.  // SunJapanShop.java
2.
3.  class SunJapanShop {
4.      public static void main(String[] args) {
5.
6.          // 配列の配列を初期化
7.          String[][] name = { {"Duke時計", "Dukeトレーナー", "Dukeペン", "Duke人形"},
8.                                {"Java One T シャツ", "Java キャップ"},
9.                                {"Sun ドライバーキット", "Sun バッグ"} };
10.
11.         System.out.println("----- 今月のお薦め商品は -----");
12.
13.         // 配列データを繰り返し処理
14.         for( int a = 0; a < name.length; a++ ){
15.             for( int b = 0; b < name[a].length; b++ ) {
16.                 System.out.println(name[a][b]);
17.             }
18.             System.out.println("***** グループ別商品出力 *****");
19.         }
20.     }
21. }
```

## 総合演習【ステップ C-5】

## Hanoi.java

```
1.  // Hanoi.java
2.
3.  class Hanoi {
4.      public static void main(String[] args) {
5.
6.          // 棒、円盤の変数を初期化
7.          char a = 'a';
8.          char b = 'b';
9.          char c = 'c';
10.         int n = 4;
11.
12.         System.out.println("円盤の枚数 4 枚");
13.
14.         // hanoi( ) メソッドの呼び出し
15.         hanoi(n, a, b, c);
16.     }
17.
18.     // hanoi( ) メソッドの定義
19.     static void hanoi(int n, char a, char b, char c) {
20.         if (n > 0) {
21.             hanoi(n-1, a, c, b);
22.             System.out.println(n + " 番目の板を " + a + " から " + b + " に移動 ");
23.             hanoi(n-1, c, b, a);
24.         }
25.     }
26. }
```



---

プログラミング・ロジック 実習課題 解答例

実習テキスト

2016 年 8 月 RevisionB.1 第 1 刷

2017 年 3 月 RevisionB.2 誤植改訂

著者 日本オラクル株式会社  
オラクルユニバーシティ

発行所 日本オラクル株式会社  
オラクルユニバーシティ

---

Printed in Japan