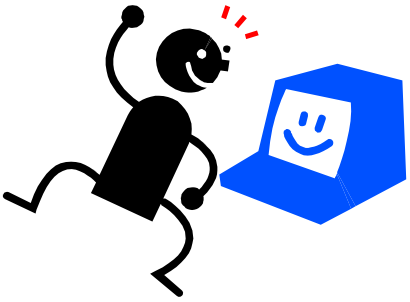
平成24年度
商業教育改善委員会報告

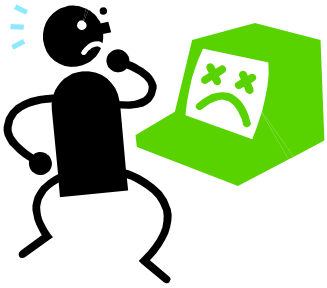
平成25年3月

兵庫県高等学校商業教育協会

「新学習指導要領におけるプログラミング」

Java 言語の演習例





平成２４年度 商業教育改善委員会

委　員　長	宿南　敏行	県立神戸商業高等学校	校　　長
副委員長	長谷川　義人	県立姫路商業高等学校	校　　長
委　員	小柴　直樹	県立神戸商業高等学校	主幹教諭
〃	山本　義史	県立神戸商業高等学校	教　　諭
〃	國澤　　剛	県立神戸商業高等学校	教　　諭
〃	細川　宗一郎	県立神戸商業高等学校	教　　諭
〃	清川　光慶	市立神港高等学校	教　　諭
〃	古川　裕士	伊丹市立高等学校	教　　諭
〃	北川　欽一	県立龍野北高等学校	教　　諭
〃	鬼頭　妙子	県立姫路商業高等学校	教　　諭
〃	福谷　千佳	県立姫路商業高等学校	教　　諭

事　務　局

奥村　裕幸	県立神戸商業高等学校	教　　諭
貞松　敏夫	県立神戸商業高等学校	教　　諭

目　　次

第１章　はじめに　…… 1

第２章　Javaに触れてみよう

(1)Javaプログラムの開発　…… 3

(2) Eclipseのセットアップ　…… 3

(3) Javaを動かしてみよう　…… 6

(4)実習を進める中で留意すること　…… 8

(5) コマンドライン（コマンドプロンプト）を利用する方法　… 10

第３章　Javaの基礎

(1)基本文法　…… 12

①変数の型

②出　　力

③入　　力

④四則演算

⑤文字データ

⑥条件文（IF文）

⑦繰り返し命令（Loop　～　While）

(2)配　　列　…… 29

(3)配列を利用した処理…… 39

①線形探索（数値型）

②線形探索（文字型）

③並べ替え

④順位付け

⑤二分探索（数値型）

第４章　総合演習　…… 56

第５章　Javaアプレットの作成　…… 64

第1章　はじめに

昭和 45 年（1970 年）からの学習指導要領により、高等学校の専門教科「商業」には「電子計算機一般」が導入された。当時、学校においても「コンピュータを学ぶ」と言えば非常に専門的な学問であり、「コンピュータについて教える」と言えば非常に多額の費用を投資して環境整備が求められた時代である。世界の先進国を手本としながら世界有数の工業国への成長を願い、わが国では人々が物質的な豊かさを求めて努力を続けてきた時代である。そのような中、商業教育に導入された情報処理教育は「時代の最先端を学び・教える」とともに「時代の最先端で活躍する」いう意味で大きな注目を浴びた。BASIC・FORTRAN・COBOL など各種プログラミング言語が登場し、情報処理教育ではアルゴリズム・プログラミング言語について学ぶ時代であった。

世界中での技術革新の中でコンピュータの小型化が進むとともに、MacOS や Windows などのオペレーティングシステムが登場し、汎用機の時代からパーソナルコンピュータの時代を迎えた。このような社会の流れの中、平成元年（1988 年）からの学習指導要領ではワープロ・表計算などのアプリケーションソフトウェアの活用教育に焦点がシフトした。以降、ネットワークの発明・進展なども関連し、情報処理教育の中心は開発者ベースの内容から利用者ベースの内容に変遷していった。いまや「情報化社会」「高度情報通信ネットワーク社会」「ユビキタス社会」など呼び方は様々であるが、ポケットから携帯情報端末があたりまえのように登場し、必要な情報を必要な時に必要に応じて得られる社会を迎えた。「スマホ」が台頭する社会となり、人々のコミュニケーション手段にも大きな変化をもたらしている。

操作性や利便性の向上は、裏を返せば「ややこしいことは知らなくていい」といったことでもある。表計算ソフト1つにしても、関数の仕組みや構造的な面に知識・関心を持たなくても使えてしまう。ただ、「アプリの使い方」を情報処理教育と考えるのは、大きな勘違いである。モノゴトの仕組みや過程・構造を知ることは、人間の能力の１つとして不可欠なものであると考える。Web ベースで社会が動き、情報という大きな無形財産が人々の生活を大きく左右する今、アルゴリズム・プログラミングの知識・能力は必須であり、その1要素が Java 言語である。

Java 言語は今、世界標準のプログラミング言語として世界中で活用されている。「プログラミングは、人々を幸せにする」－ぜひ商業教育の中で科目「プログラミング」だけでなく「電子商取引」「課題研究」などでも改めてプログラミングを学ぶ・教えることの専門性を育んで頂きたい。「過去を知れば、未来が見える」。いまどきのプログラミングを、ぜひすべての商業教育の中で導入・発展して頂けることを、強く願う。

<table>
<tr><th></th><th>昭和 45 年</th><th>昭和 53 年</th><th>平成元年</th><th>平成 11 年</th><th>新
（平成 25 年～）</th></tr>
<tr><td>インターネット
マルチメディア</td><td></td><td></td><td></td><td rowspan="2">文書デザイン</td><td rowspan="3">電子商取引</td></tr>
<tr><td rowspan="2">文書情報</td><td></td><td></td><td>文書処理</td></tr>
<tr><td></td><td></td><td rowspan="4">情報処理</td><td>商業技術</td></tr>
<tr><td>ソフトの活用</td><td></td><td></td><td rowspan="2">情報処理</td><td rowspan="2">情報処理</td></tr>
<tr><td>概論・基礎理解
ハード・ソフト</td><td rowspan="2">電子計算機
一般</td><td rowspan="3">情報処理Ⅰ</td></tr>
<tr><td rowspan="2">プログラミング基礎</td><td>プログラミング</td><td>プログラミング</td></tr>
<tr><td>プログラミングⅠ</td><td rowspan="3">情報処理Ⅱ</td><td rowspan="2">プログラミング</td><td></td><td></td></tr>
<tr><td>プログラミング応用</td><td rowspan="2">プログラミングⅡ</td><td></td><td></td></tr>
<tr><td>システム開発</td><td>情報管理</td><td rowspan="3">ビジネス情報</td><td>ビジネス情報管理</td></tr>
<tr><td>OR</td><td>経営数学</td><td>経営数学</td><td rowspan="2">経営情報</td><td rowspan="2">ビジネス情報</td></tr>
<tr><td>ソフトの応用活用</td><td></td><td></td></tr>
</table>

年 間 指 導 計 画 案

科　目:プログラミング　単位数:2～4単位

1　目　標

プログラミングに関する知識と技術を習得させ、プログラムの役割や重要性について理解させる。
ビジネスの諸活動においてコンピュータを合理的に活用する能力と態度を育てる。

2　年間計画

学期	月	学習内容	学習のねらい・目標	備考
1	4	ハードウェア ア　データの表現 イ　中央処理装置 ウ　主記憶装置 エ　周辺装置	コンピュータ内部におけるデータの表現及びコンピュータが動作する仕組みなどを取り扱い、ハードウェアの概要について理解させる	2進数で動くコンピュータの特性を理解する ハードウェアについての知識を身につける
		ソフトウェア ア　システムソフトウェア イ　応用ソフトウェア	役割や機能を取り扱い、ソフトウェアの概要について理解させる データ処理の形態及びプログラム言語の種類と特徴を取り扱い、データ処理とプログラムの概要について理解させる	ワープロソフト、表計算ソフトなど具体例をあげて説明する。
	5	コンピュータとプログラミング ア　ビジネスとデータ処理 イ　プログラム言語の種類と特徴	プログラムを作成するための基礎的な知識と技術を習得させる	流れ図記号の学習 Java の基本文法の学習
	6	プログラミングの基礎 ア　プログラミングの手順 イ　アルゴリズムの表現技法 ウ　データ構造と制御構造 エ　変数と定数		コンピュータを使った実習開始 『Eclipse』の基本操作について説明する
	7	オ　データの入出力と演算 カ　条件判定と繰り返し処理		問題分析を行い、自らプログラムを組んでいくことで深く理解し技術を習得する
2	9	オブジェクト指向の考え方 プログラミングの応用 ア　配列の利用 イ　ファイル入出力	クラスの定義とオブジェクトの生成 配列を利用する意義について理解させるとともに、配列を利用して、集計、探索、順位づけ及び内部整列を行うための技法を習得させる	(情報処理検定試験) 配列の意義を、配列を使わない場合と比較することで確認する
	10			
	11			
	12			
3	1	ウ　メソッドの利用 エ　ユーザインターフェース オ　Applet の利用	クラス、メソッドについて理解させるとともに、クラス、メソッドを利用するための技法を習得させる アプレットの Web 利用方法について、実習を通じて理解を深める。	(情報処理検定試験) クラスとメソッドについて確認する ビジネスでの活用を考察する
	2			
	3	文字や画像などの処理		

第2章　Java に触れてみよう

（1）Java プログラムの開発

①フリーソフトウェア「Eclipse」を利用する方法

「プログラミングって難しそう…」このような不安は、誰でも持ってしまうものです。たしかに、プロになると、言語独自の専門的な特性や文法・手法を理解し、それらを組み合わせてプログラムを作成・開発していくという能力が必要になるわけですが、この際にプログラミング以外のところでの知識が多く必要になり、つまずいてしまう場合があります。しかし、初心者がプログラミング言語を学ぶにあたっては、そのようなレベルを目指す必要はありません。

私たちが英語や中国語など他の国の言葉を読み書きできるようになりたいと志す際を思い出してみてください。文法や単語の存在は意識しながらも、なんとなくわかれば、ボディランゲージなどを組み合わせることにより、「それなりな会話」はできるようになりますよね。この経験とプログラミング言語に触れるというのはよく似ていて、私たちとコンピュータのコミュニケーションを助けてくれるツールの1つが、以下に示す「Eclipse（エクリプス）」です。

Eclipse には、以下の特徴があります。

- ·GUI（Graphical User Interface）ツールなので、コマンド（専門的な命令）を覚えなくても使える
- ·1 つのウィンドウ内で、プログラミングに必要な作業ができる
- ·プログラム開発を効率よく行うための支援機能が多い
- ·多くの Java プログラマに人気があり、使っている人が多い

Eclipse は無償で入手できる開発環境の1つで、単純な画面操作でプログラムの編集・コンパイル（翻訳）・実行ができるソフトです。デバッグ機能が充実しているので、Eclipse を使えばそれなりな会話ができるようになり、初心者でもプログラミングに集中できる…というわけです。

②コマンドライン（コマンドプロンプト）を利用する方法

Windows であればメモ帳などのテキストエディタと Java のJDK（Java 開発キット）があれば、コマンドライン（コマンドプロンプト）を利用して、コンパイル・実行ができます。

（2）Eclipse のセットアップ

Eclipse の開発環境に必要なものは、「Eclipse 本体」と「それを日本語化するパッケージ」です。All in One でダウンロードできるサイトがあり、ここからまとめてダウンロードすれば、保存していくフォルダを用意するだけですぐに利用することが出来ます。（機能も網羅している分、「All in One でのダウンロード」は容量が大きくなってしまいますが、細かな機能選択等が不要になりますので、まとめて一括の入手をお勧めします。）

では、Eclipse のダウンロードから起動、そして初めてのプログラミングまで進めてみましょう。

①Eclipse のダウンロード

右のサイトにアクセスします。 http://mergedoc.sourceforge.jp/

使用するバージョンを選びます。(現在最新は[4.2Juno]になります。最新の Juno は、現状では安定性が低いため【2012年末現在】、本冊子では[3.7Indigo]をダウンロードします。)

なお、どのバージョンでも本冊子で扱うプログラムの動作は可能ですから、ダウンロードランキングを参考に、その時に一番使われているバージョンを選択するとよいでしょう。

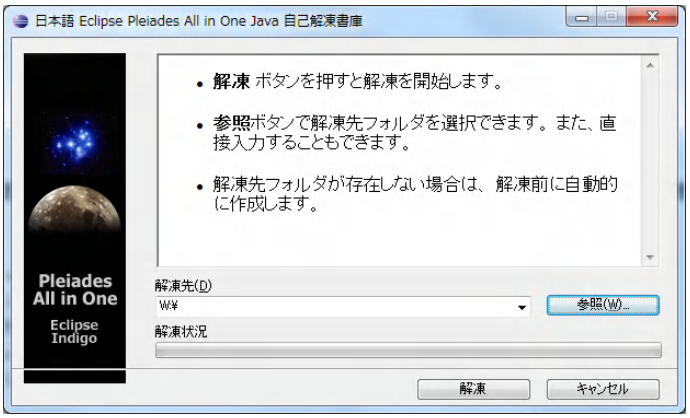
「Java の JRE あり」版を選んで、ダウンロードします。

②Eclipse のインストール

任意のフォルダに解凍します。フォルダ名は「d:¥eclipse」など短いフォルダ名にしてください。

※ 日本語フォルダ名や「c:¥Program Files」などフォルダ名に半角スペースがあると、動作しない場合があります。

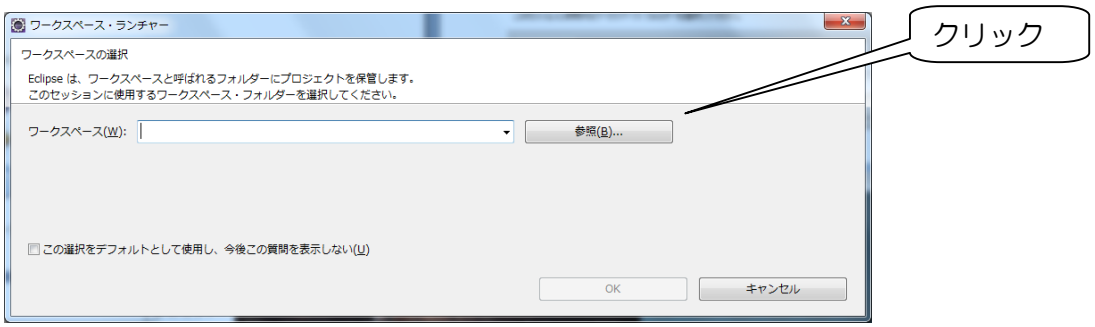
※ ネットワークドライブやUSBメモリなどに解凍しても動作します。



③Eclipse の起動

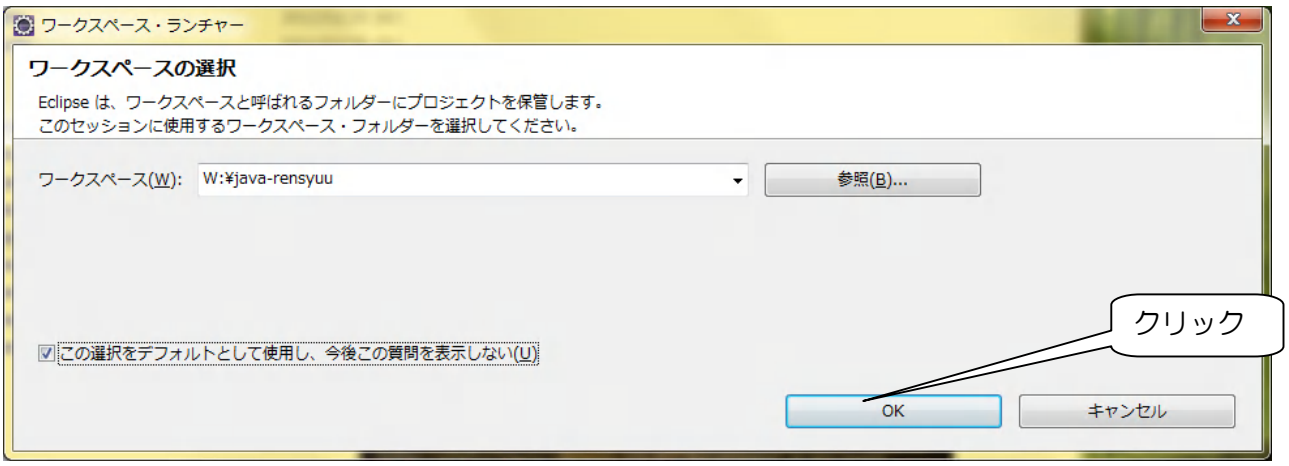
・解凍したフォルダの中で「eclipse.exe」の丸いアイコンをダブルクリック

・ワークスペースの選択をします。ワークスペースはフォルダの保存場所の指定にあたります。
参照ボタンをクリック

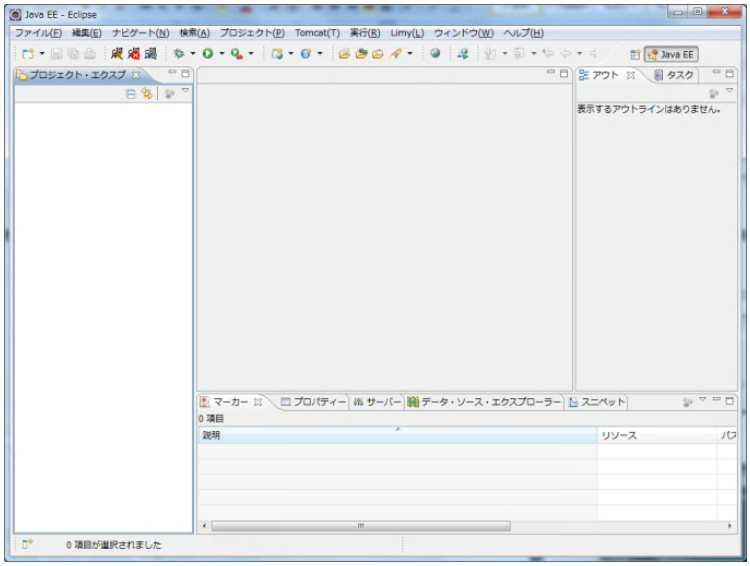


・保存する場所を指定します。ネットワークドライブやUSBメモリでも可能です。

例 ネットワークドライブ（今回は w ドライブ）内に作った「java-rensyuu」フォルダを選択しOKをクリック
（あらかじめ「java-rensyuu」フォルダを作成しています。）



Eclipse が起動しました。



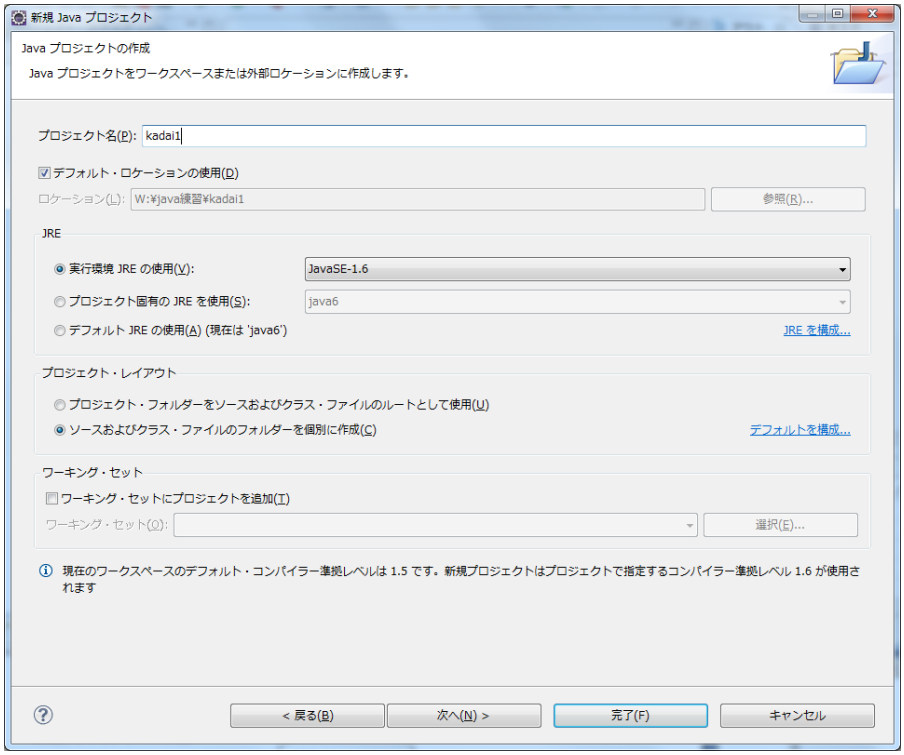
（3）Java を動かしてみよう

例題　画面上に「Hello, world!」と表示するプログラムを作成しなさい。 プロジェクト名：kadai1 ファイル名：Hello.java

```
public class Hello{
        public static void main(String[] args) {
                System.out.println("Hello, world!");
        }
}
```

①［ファイル］→［新規］→［その他］→［Javaプロジェクト］→［次へ］と順に選択

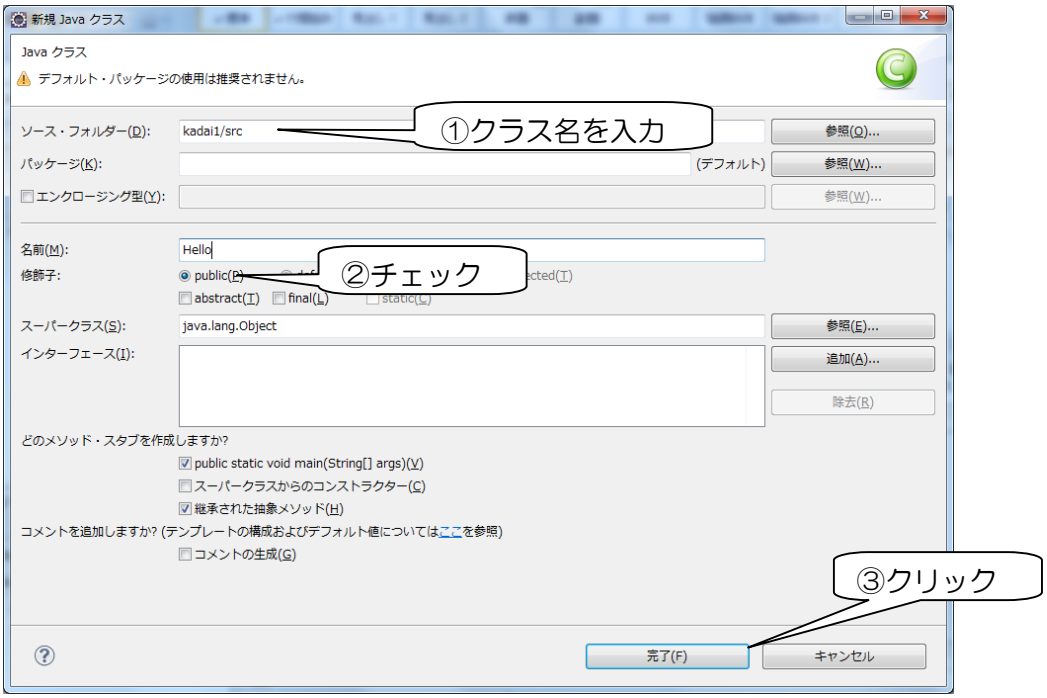
②プロジェクト名【kadai1】を入力し、［完了］ボタンをクリック



③「パースペクティブを開きますか？」と聞いてきたら、［はい］をクリック

④新規Javaクラスの作成

クラス名を入力し、public static void main(String[] args)にチェックを入れる。



⑤コードが自動生成されるので、不要な部分を削除し、図のようにプログラムを入力して実行ボタンをクリックします。

⑥保管して起動　OK　ボタンをクリック

⑦実行結果が表示されます。

　このような流れで課題を作成していきます。プログラムの中身以外の操作は共通ですので、以後の課題では操作については省略していきます。

【課題例のプログラムの説明】

```
public class Hello {
	public static void main(String[] args) {
		System.out.println("Hello,world!");
	}
}
```

このプログラムの内容は文字列の表示です。Java言語のプログラムで文字を表示するには、「Println()」というメソッド(一連の処理の集まり)を使います。これらの前には「System.out」をつけます。

```
System.out.println("Hello,world!");
```

(4)実習を進める中で留意すること

次章より課題を進めていきますが、はじめのうちは以下のようなミスケースがよく起こると思います。

①クラス名が違う。または最初が大文字でない。(java の文法では大文字と小文字の区別をします)

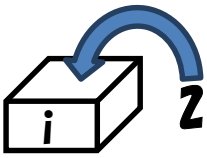
第3章　Java の基礎

(1)基本文法

①変数の型

【変数の宣言】　　型　識別子；　　例　int　hensu;

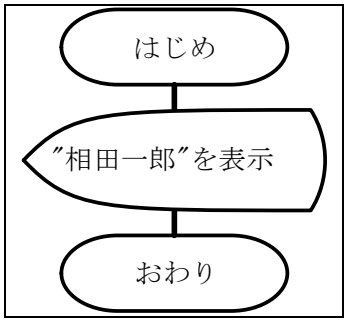
【変数への値の代入】　　変数 = 値；　　例　int　i
　　i　=　2;



②出力

設問②－1　自分の名前を表示しなさい。

★流れ図



★プログラム例　　ファイル名：Hyoji.java

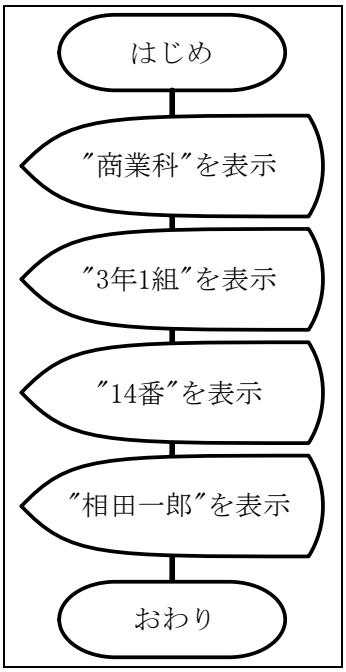
```
1 public class Hyoji{
2         public static void main(String[] args){
3                 System.out.println("相田一郎");
4         }
5 }
```

★実行結果

```
相田一郎
```

設問②－2　学科、学年、クラス、番号、名前を表示しなさい。

★流れ図



★プログラム例　　ファイル名：Hyoji2.java

```
1  public class Hyoji2{
2          public static void main(String[] args){
3                  System.out.println("商業科");
4                  System.out.println("3年1組");
5                  System.out.println("14番");
6                  System.out.println("相田一郎");
7          }
8  }
```

★実行結果

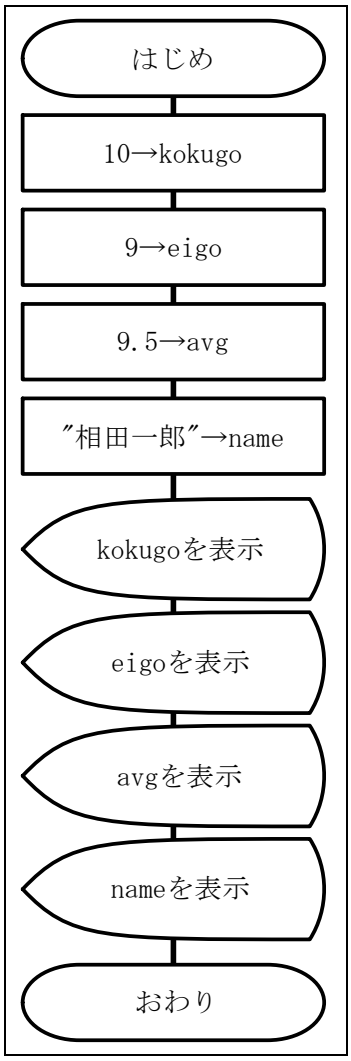
```
商業科
3年1組
14番
相田一郎
```

③変数を使った処理

設問③－１　国語、英語の小テストの点数、平均点、名前を表示しなさい。

２教科の点数、平均点、名前は変数に入れること。
平均点は小数で表示すること。

★*流れ図*



★プログラム例　　ファイル名：Hensu.java

```
1  public class Hensu{
2          public static void main(String[] args){
3                  // 変数宣言
4                  int    kokugo;
5                  int    eigo;
6                  double avg;
7                  String name;
8                  // 値代入
9                  kokugo = 10;
10                 eigo = 9;
11                 avg = 9.5;
12                 name = "相田一郎" ;
13                 // 表示
14                 System.out.println("国語："+kokugo);
15                 System.out.println("英語："+eigo);
16                 System.out.println("平均："+avg);
17                 System.out.println("名前："+name);
18         }
19 }
```

★実行結果

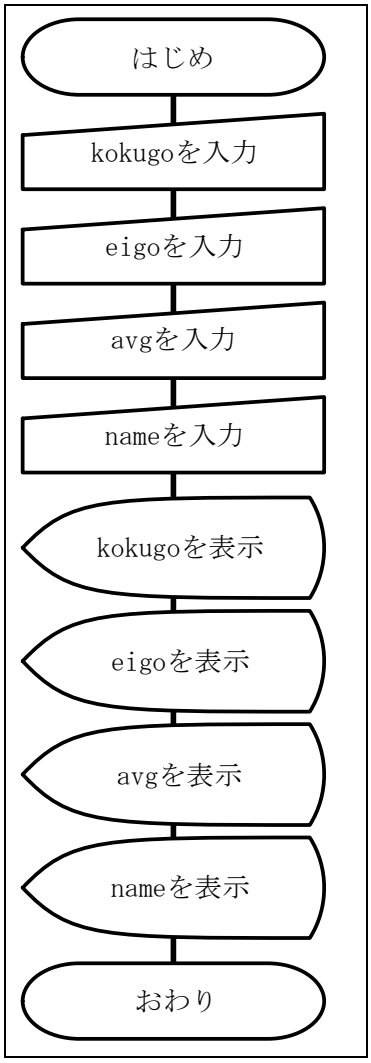
```
国語：10
英語：9
平均：9.5
名前：相田一郎
```

④入力

設問④－１　「国語の点数」「英語の点数」「平均点」「名前」を入力し、その内容をそれぞれ表示しなさい。

２教科の点数、平均点、名前はそれぞれ変数に入れること。
平均点は小数で表示すること。

★*流れ図*



★プログラム例　　ファイル名：Nyuryoku.java

```
1  import java.util.*;
2  public class Nyuryoku{
3          public static void main(String[] args){
4                  // 変数宣言
5                  int    kokugo;
6                  int    eigo;
7                  double avg;
8                  String name;
9                  // 値代入
10                 Scanner scan = new Scanner(System.in);
11                 System.out.println("国語の点数を入力してください");
12                 kokugo = scan.nextInt();
13                 System.out.println("英語の点数を入力してください");
14                 eigo   = scan.nextInt();
15                 System.out.println("平均点を入力してください");
16                 avg    = scan.nextDouble();
17                 System.out.println("名前を入力してください");
18                 name   = scan.next();
19                 // 表示
20                 System.out.println("国語："+kokugo);
21                 System.out.println("英語："+eigo);
22                 System.out.println("平均："+avg);
23                 System.out.println("名前："+name);
24         }
25 }
```

★実行結果

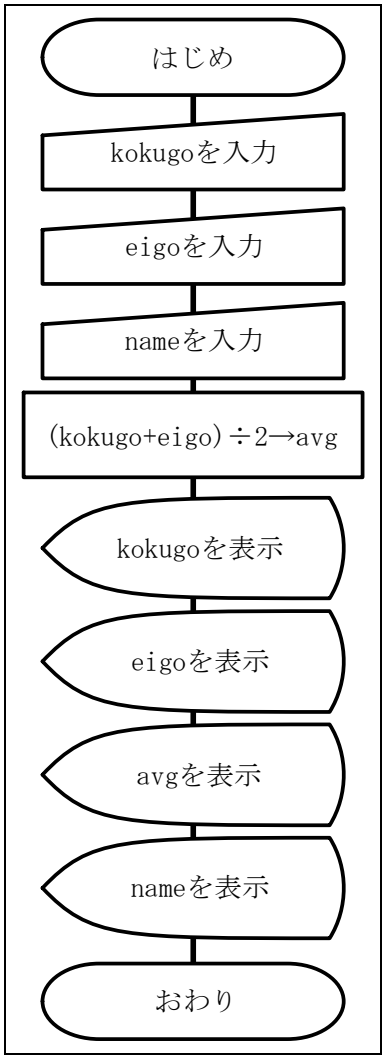
```
国語の点数を入力してください
10
英語の点数を入力してください
9
平均点を入力してください
9.5
名前を入力してください
相田一郎
国語：10
英語：9
平均：9.5
名前：相田一郎
```

⑤四則演算

設問⑤－１　「国語の点数」「英語の点数」「名前」を入力し、２教科の平均点を求めて入力した内容とともにそれぞれ表示しなさい。

２教科の点数、名前、平均点はそれぞれ変数に入れること。
平均点は小数で表示すること。

★*流れ図*



★プログラム例　　ファイル名：Shisoku.java

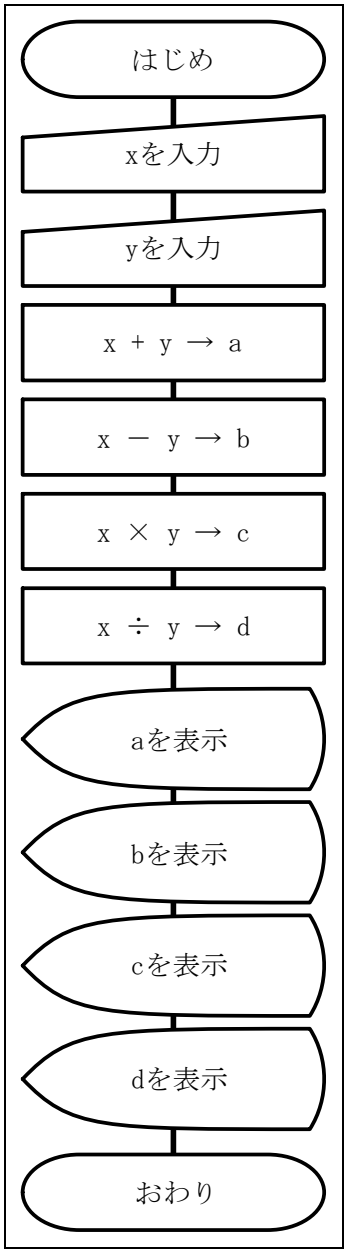
```
1  Import  java.util.*;
2  public class Shisoku{
3          public static void main(String[] args){
4                  // 変数宣言
5                  int    kokugo;
6                  int    eigo;
7                  double avg;
8                  String name;
9                  // 値代入
10                 Scanner scan = new Scanner(System.in);
11                 System.out.println("国語の点数を入力してください");
12                 kokugo = scan.nextInt();
13                 System.out.println("英語の点数を入力してください");
14                 eigo   = scan.nextInt();
15                 System.out.println("名前を入力してください");
16                 name   = scan.next();
17                 // 平均点計算
18                 avg  = ( kokugo + eigo ) / 2.0;
19                 // 表示
20                 System.out.println("国語："+kokugo);
21                 System.out.println("英語："+eigo);
22                 System.out.println("平均："+avg);
23                 System.out.println("名前："+name);
24         }
25 }
```

★実行結果

```
国語の点数を入力してください
10
英語の点数を入力してください
9
名前を入力してください
相田一郎
国語：10
英語：9
平均：9.5
名前：相田一郎
```

設問⑤－２　２つの数値を入力し、和、差、積、商を求めてそれぞれ表示しなさい。

★*流れ図*



★プログラム例　　ファイル名：Shisoku2.java

```
1  import java.util.*;
2  public class Shisoku2 {
3          public static void main(String[] args){
4                  // 変数宣言
5                  int x,y,a,b,c,d;
6                  // 入力
7                  Scanner scan = new Scanner(System.in);
8                  System.out.println("数値を入力してください");
9                  x = scan.nextInt();
10                 System.out.println("数値を入力してください");
11                 y   = scan.nextInt();
12                 // 計算
13                 a  = x + y;
14                 b  = x - y;
15                 c  = x * y;
16                 d  = x / y;
17                 // 表示
18                 System.out.println("和："+a);
19                 System.out.println("差："+b);
20                 System.out.println("積："+c);
21                 System.out.println("商："+d);
22         }
23 }
```

★実行結果

```
数値を入力してください
6
数値を入力してください
2
和：8
差：4
積：12
商：3
```

※参考：次のような演算子の使い方もあります。

単項演算子	変換前		変換後
+=	a = a + b	→	a += b
-=	a = a - b	→	a -= b
*=	a = a * b	→	a *= b
/=	a = a / b	→	a /= b

インクリメント演算子　++

　　i ++ は　i = i + 1 の計算式と同じ演算をします。

デクリメント演算子　－

　　i － は　i = i － 1 のの計算式と同じ演算をします。

⑥文字列データ

設問⑥－１　名前をキーボードから入力し、実行結果のように表示するプログラムを作成しなさい。

入力データ

名前
na

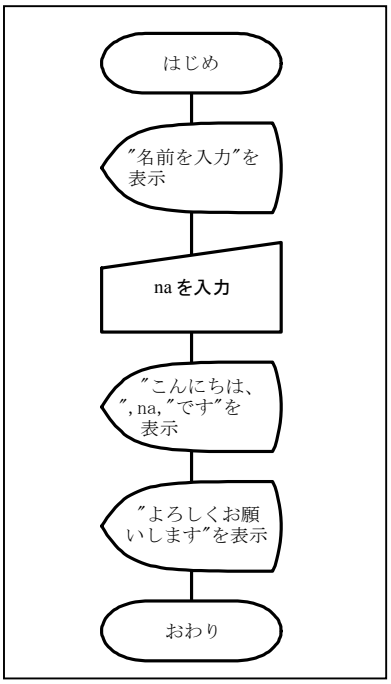
実行結果

名前を入力
花子
こんにちは、花子です
よろしくお願いします。

処理条件

1. キーボードから名前を入力し、あいさつ文を表示する。

★*流れ図*



★*プログラム*

```
1  import java.util.*;
2  public class Aisatu {
3      public static void main(String args []){
4          String na ;
5          Scanner scan = new Scanner(System.in);
6          System.out.println("名前を入力");
7          na = scan.next();
8          System.out.println("こんにちは、" + na + "です");
9          System.out.println("よろしくお願いします。");
10     }
11 }
12
```

⑦条件文(IF 文)

設問⑦－1　区分と人数をキーボードから入力し、実行結果のように表示するプログラムを作成しなさい。

入力データ

区分 ku	人数 nin

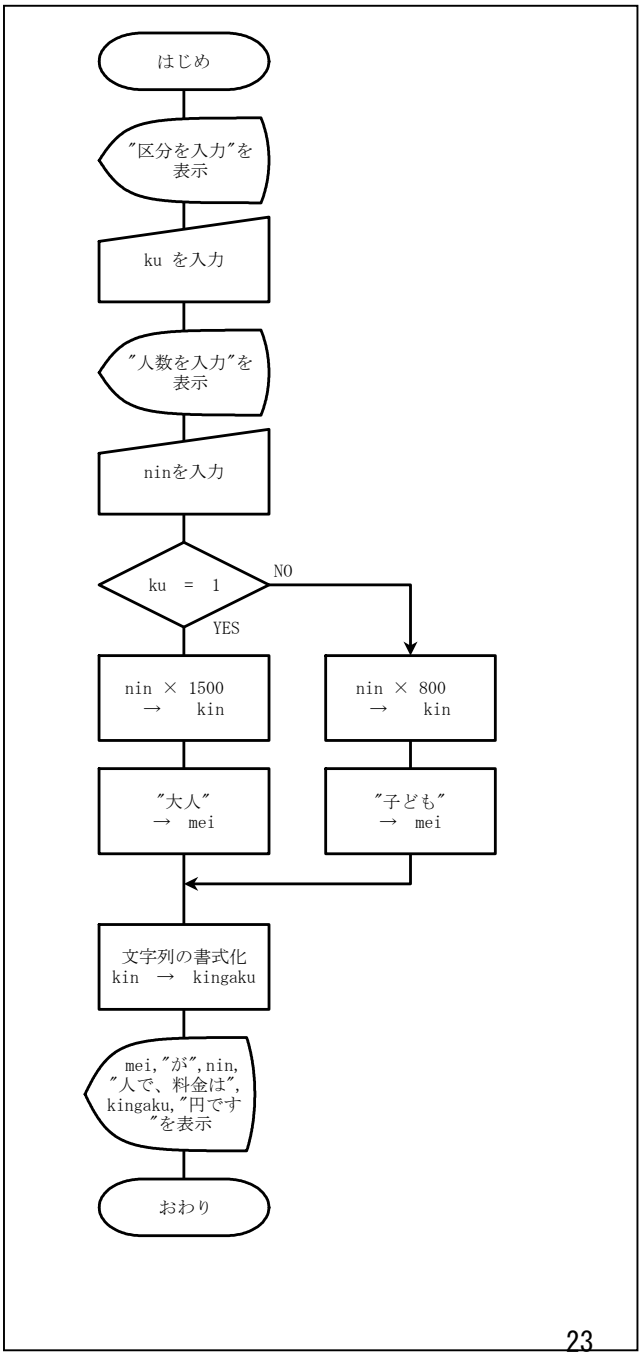
実行結果

区分を入力
1
人数を入力
2
大人が2人で、料金は　3,000円です

処理条件

1. キーボードから区分と人数を入力し、入場料金を求め表示する。
2. 区分は、大人が 1、子どもが 2 である。
3. 料金は、大人が 1500、子どもが 800 とする。

★*流れ図*



★プログラム

```
1  import java.util.*;
2  publicclass kubun{
3      publicstaticvoid main(String args []) {
4          int ku, nin, kin;
5          String mei;
6          Scanner scan = new Scanner(System.in);
7          System.out.println("区分を入力");
8          ku = scan.nextInt();
9          System.out.println("人数を入力");
10         nin = scan.nextInt();
11         if (ku == 1 ){
12            kin = nin * 1500;
13            mei = "大人";
14         }
15         else{
16            kin = nin * 800;
17            mei = "子ども";
18         }
19
20         String kingaku = String.format("%,7d", kin);   //  ※1
21          //Stringクラスのformat()メソッドで文字列を書式化する
22
23         System.out.println(mei + "が" + nin + "人で、"+ "料金は" + kingaku + "円です");
24       }
25 }
```

※1　String.format("%,7d", kin)10 進数形式の整数 7 桁で右寄せコンマ付き

⑧繰り返し命令

設問⑧－1　２から 100 までの偶数の合計を求め、実行結果のように表示するプログラムを作成しなさい。

処理条件

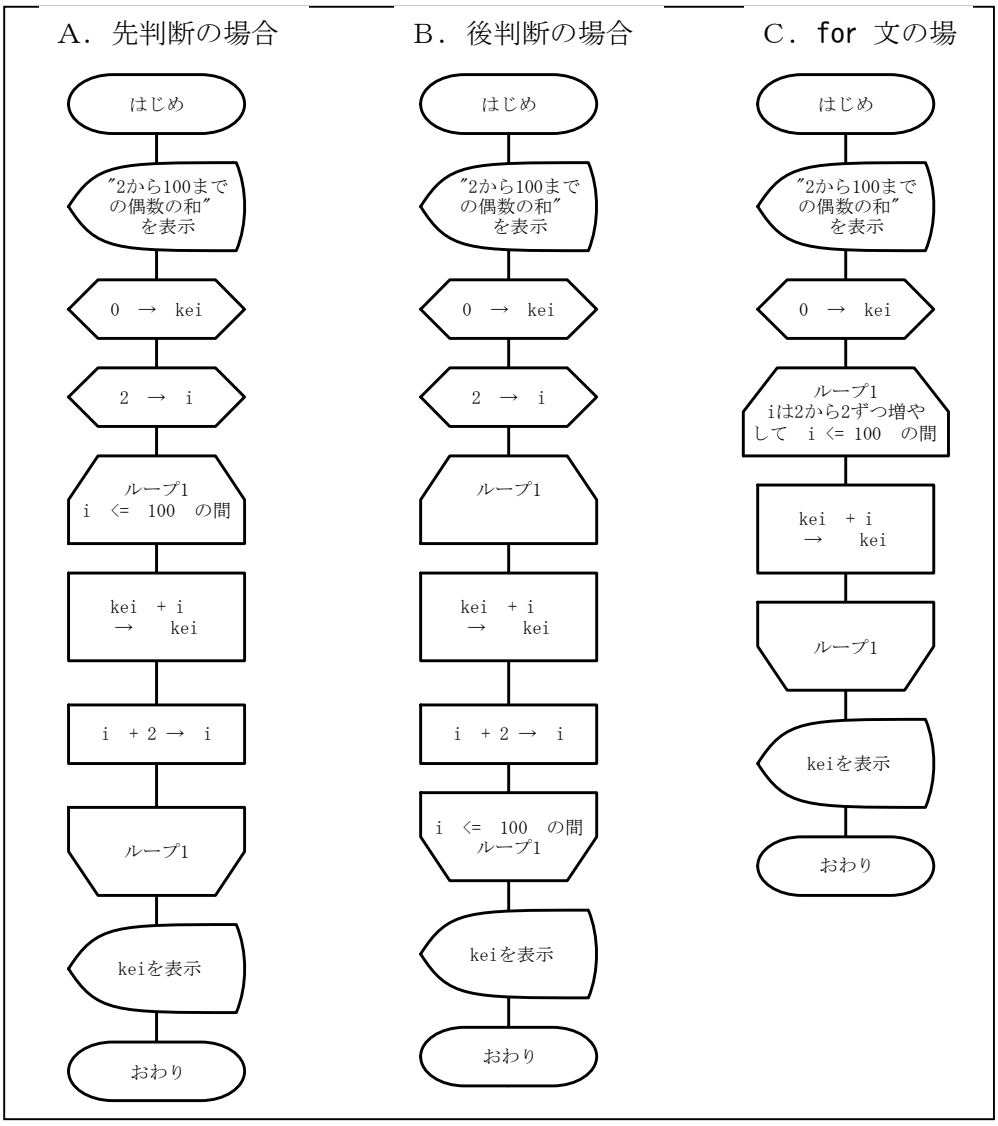
2 から 100 までの偶数の合計を求めて表示する。

実行結果

```
2 から 100 までの偶数の和
2550
```

※繰り返し命令には、以下の３パターンがあります。

A【先判断】	while(条件式)　{ 処理 }	条件式が成立する間、処理を繰り返す。
B【後判断】	do { 処理 }　while(条件式)；	処理を実行した後に条件の判定を行い、条件が成立する間、処理を繰り返す。
C【for 文】	for（初期化式；条件式；更新式）{ 処理 }	変数の値を変化させて条件式が成立するまで処理を繰り返す。
	初期化式……変数の最初の値を設定する　例　int i=2 条件式……繰り返しの条件を記述する 更新式……処理の実行が一巡した時に実行する それぞれ省略は可能ですが、初期化式以外を省略すると無限ループになります。	



A．while(先判断)文の場合

```
1  publicclass gusu {
2      publicstaticvoid main(String args []){
3          int kei ;
4          int i ;
5          System.out.println("2から100までの偶数の和");
6          kei = 0;
7          i = 2 ;
8
9           //繰り返し処理 前判定
10         while (i <= 100){  //  iが100以下のあいだ繰り返す
11          kei = kei + i;
12          i = i + 2;
13         }
14         System.out.println(kei);
15       }
16 }
```

B．do　while(後判断)文の場合

```
1  publicclass gusu {
2      publicstaticvoid main(String args []){
3          int kei ;
4          int i ;
5          System.out.println("2から100までの偶数の和");
6          kei = 0;
7          i = 2 ;
8
9           //繰り返し処理 後判定
10         do{
11          kei = kei + i;
12          i = i + 2;
13         } while (i <= 100); //  iが100以下のあいだ繰り返す
14         System.out.println(kei);
15       }
16 }
```

C．for 文の場合

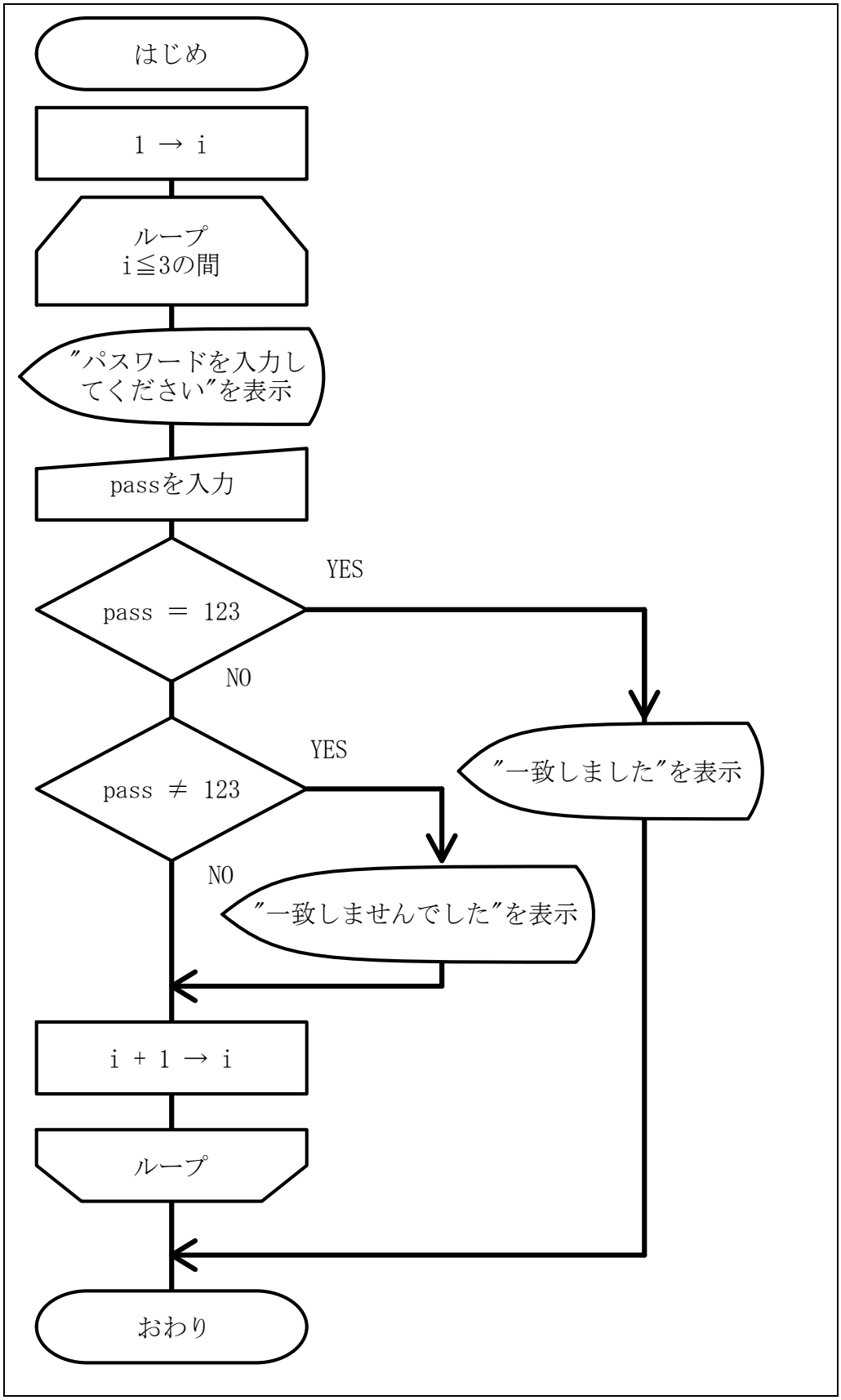
```
1  publicclass gusu3 {
2      publicstaticvoid main(String args []){
3          int kei ;
4          int i ;
5          System.out.println("2から100までの偶数の和");
6          kei = 0;
7
8          //繰り返し処理
9          for(i = 2; i <= 100; i = i + 2){
10             kei = kei + i;
11         }
12         System.out.println(kei);
13       }
14 }
```

設問⑧－２　次のプログラムを作成しなさい。

・パスワードを入力する。パスワードは数値のみ。

・パスワードチェックは３回までとする。

★*流れ図*



★プログラム例　　ファイル名：Password.java

```
1  importjava.util.*;
2  public class Password{
3      public static void main(String[] args) {
4          // 変数宣言
5          int pass;
6          // 入力
7          Scanner scan = new Scanner(System.in);
8          int i = 1;
9          while ( i<=3 ) {
10             //キーボードからの入力
11             System.out.print("パスワードを入力して下さい：");
12             pass = scan.nextInt();
13             //  チェック
14             if (pass  == 123 ) {
15                 System.out.println("一致しました");
16                 //ループ脱出
17                 break;
18             }
19             if (pass  != 123 ) {
20                 System.out.println("一致しませんでした");
21             }
22             i++;
23         }
24     }
25 }
```

★実行結果

```
パスワードを入力してください：111
一致しませんでした
パスワードを入力してください：112
一致しませんでした
パスワードを入力してください：123
一致しました
```

※インクリメント演算　i++

i=i+1 の計算式と同じです

(2)配列

設問１　兵庫商事の売上データをキーボードから入力し、実行結果のように表示するプログラムを作成しなさい。

入力データ

品番	数量
ban	su

実行結果

```
品番を入力してください 1
数量を入力してください 10
  ⋮
品番      数量計      割合(%)
1         1018.2
2         2036.4
3         25          45.5
合計      55
平均      18.3
```

処理条件

1. キーボードから品番と数量を入力し、品番ごとに数量計を配列 tsu に求める。また、合計を tsu(5) に求める。

   配列 tsu

(0)	(1)	(2)	～	(5)

2. 品番は 1～5 の 5 種類である。
3. 品番に 0 が入力されたら入力を終了し、実行結果のように割合を求め、品番と数量合計と割合を表示する。
4. 最後に合計と平均を表示する。
5. 1～5 以外の品番が入力されたら、品番エラーと表示する。
6. 売上データのない品番は表示しない。

※型変換…型の違う値同士の演算

(1)暗黙的な型変換

サイズの小さい型から大きな型に自動的に変換する。

例）int 型 → long 型

サイズの大きな型から小さな型に変換すると、エラーになる。

(2)キャスト演算子

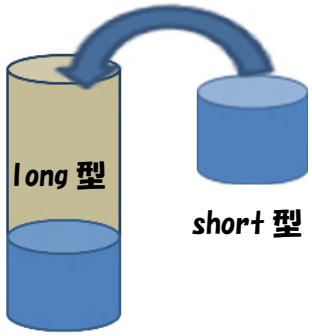
明示的に型変換を行うキャスト演算子を用いれば、サイズの大きな型から小さな型に変換できる。

例）整数同士の割り算の結果を小数で求めることができる。

【異なる型の変数間での変換】

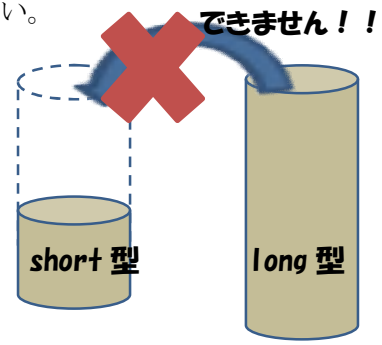
「long 型変数　=　short 型変数」

何の問題もなく行われる。

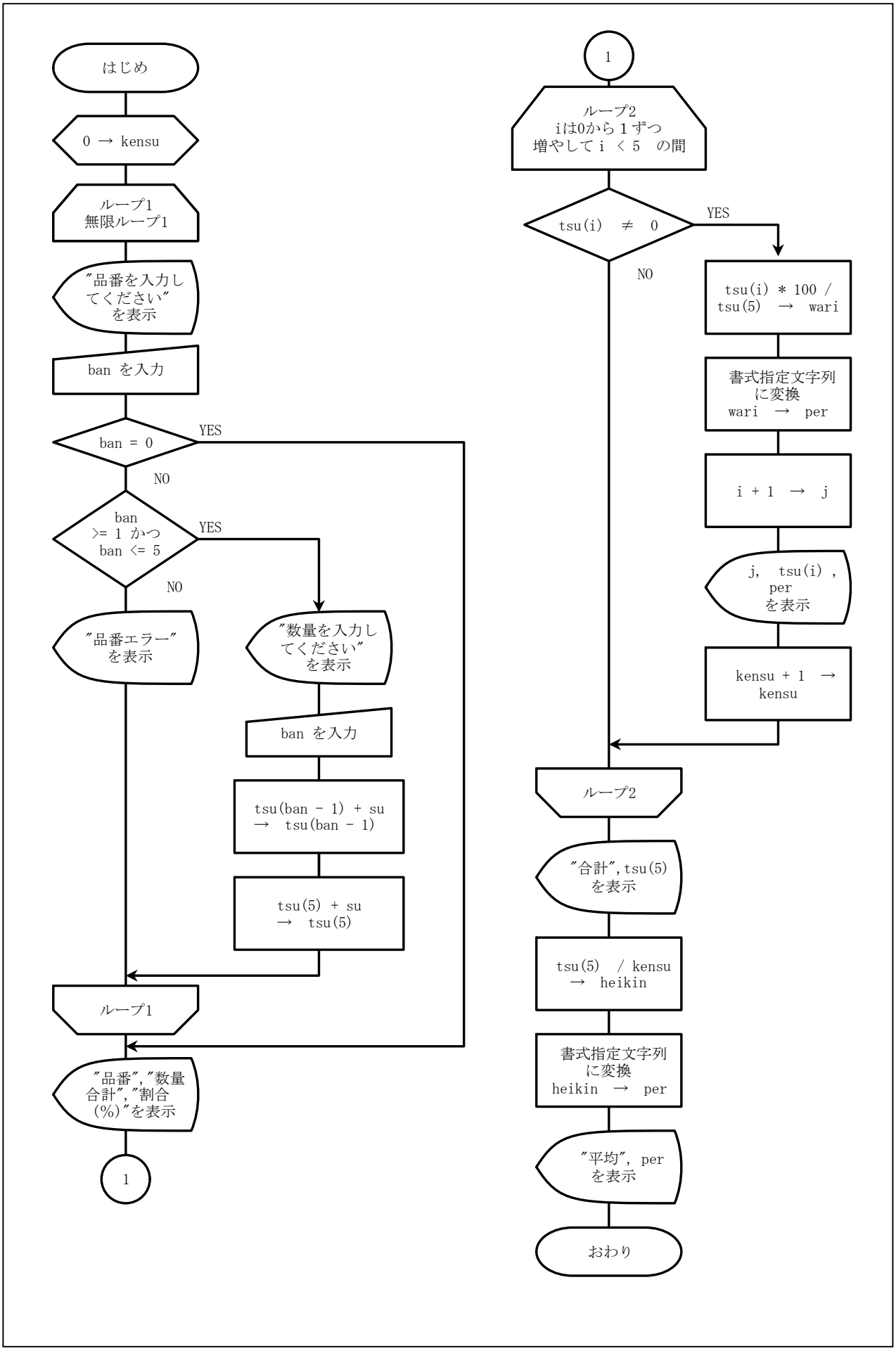


「short 型変数　=　long 型変数」

格納先が小さいので long 型の一部しか格納できない。



★流れ図


はじめ
0 → kensu
ループ1
無限ループ1
"品番を入力してください"を表示
ban を入力
ban = 0
YES
NO
ban >= 1 かつ ban <= 5
YES
NO
"品番エラー"を表示
"数量を入力してください"を表示
ban を入力
tsu(ban - 1) + su → tsu(ban - 1)
tsu(5) + su → tsu(5)
ループ1
"品番","数量合計","割合(%)"を表示
1
1
ループ2
iは0から1ずつ増やしてi < 5 の間
tsu(i) ≠ 0
YES
NO
tsu(i) * 100 / tsu(5) → wari
書式指定文字列に変換
wari → per
i + 1 → j
j, tsu(i), per を表示
kensu + 1 → kensu
ループ2
"合計",tsu(5)を表示
tsu(5) / kensu → heikin
書式指定文字列に変換
heikin → per
"平均", per を表示
おわり

★プログラム

```
1  import java.util.*;
2  public class Wariai {
3      public static void main(String[] args){
4              int ban, su;
5              int i, j;
6              double wari,  heikin;
7              int kensu = 0;
8              int[ ] tsu = new int[6];
9              Scanner scan = new Scanner(System.in);
10
11             for( ; ; ){  //無限ループ
12                 System.out.print("品番を入力してください");
13                 ban = scan.nextInt();
14                 if(ban == 0){    //ban が 0 と等しければ
15                     break;  //無限ループを抜け出す
16                 }
17                 if ( ban >= 1 && ban <= 5 ){ //ban が 1 以上でかつ５以下
18                     System.out.print("数量を入力してください");
19                     su= scan.nextInt();
20                     tsu[ ban - 1] = tsu[ ban - 1] + su ;
21                     tsu[5] = tsu[5] + su ;
22                 }
23                 else {
24                     System.out.println("品番エラー");
25                 }
26             }
27             System.out.println("品番 " + "¥t" + "数量合計" + "¥t" + "割合(%)"); //※1
28
29             for(i = 0;i < 5; i++){
30                 if(tsu[i] != 0){  // tsu[i] が 0  と等しくなければ
31                     wari = (double)(tsu[i]) * 100 / tsu[5];//  ※2 キャスト演算子
32                     String per = String.format("%.1f", wari);// ※3
33                     j = i + 1;
34                     System.out.println( j + "¥t" + tsu[i] + "¥t" +per);
35                     kensu++;  //kensu = kensu + 1
36                 }
37             }
38             System.out.println("合計" + "¥t" + tsu[5]);
39             heikin = (double)(tsu[5]) / kensu;
40             String per = String.format("%.1f", heikin);
41             System.out.println("平均 " + "¥t" + per);
42         }
43     }
```

※1"¥t"タブコード

※2　heikin = (double)(tsu[5]) / kensu
　　キャスト演算子　　例）　実数データ名　＝　(double)（整数データ名）　÷　整数データ名

※3String.format("%.1f", wari)　浮動小数点数を小数１桁まで表示

設問２　兵庫スポーツ株式会社のある１週間の売上データをキーボードから入力し、
実行結果のように表示するプログラムを作成しなさい。

入力データ

店番	品番	金額
ten	hin	kin

実行結果

```
店番を入力して下さい 4
品番を入力して下さい 1
金額を入力して下さい 300
   ∫
                    店別品別売上金額一覧
 神戸支店  野球：  1,000   サッカー：  2,500   ～  テニス：  1,500  計:  24,000
 姫路支店  野球：  2,000   サッカー：  1,100   ～  テニス：  1,800  計:  21,000
 赤穂支店  野球：  2,400   サッカー：  2,100   ～  テニス：  2,500  計:  19,000
 城崎支店  野球：  1,200   サッカー：  2,800   ～  テニス：  1,600  計:  17,000
 三田支店  野球：  2,300   サッカー：  2,600   ～  テニス：    900  計:  19,500
 計        野球：11,200   サッカー：12,500   ～  テニス：10,600  計: 122,500
```

処理条件

1. キーボードから店番と品番と金額を入力し、店番別品番別金額合計を配列 tkin に求める。

   配列 tkin

	(0)	(1)	(2)	(3)	(4)
(0)					
(1)					
∫					
(5)					

2. 店番は 1 が神戸、2 が姫路、3 が赤穂、4 が城崎、5 が三田の 5 種類であり、配列 tmei に記憶する。

   配列 tmei

(0)	(1)	(2)	(3)	(4)
神戸	姫路	赤穂	城崎	三田

3. 品番は 1 が野球、2 がサッカー、3 が陸上、4 がテニスの 4 種類であり、配列 hmei に記憶する。

   配列 hmei

(0)	(1)	(2)	(3)	(4)
野球	サッカー	陸上	テニス	計

4. 1～5 以外の店番が入力されたら、店番エラーと表示する。
5. 1～4 以外の品番が入力されたら、品番エラーと表示する。
6. 店番に 0 が入力されたら入力を終了し、実行結果のように店番別品番別金額合計、店別計、品番別計を表示する。

この問題では、次の３つの処理パターンがある。

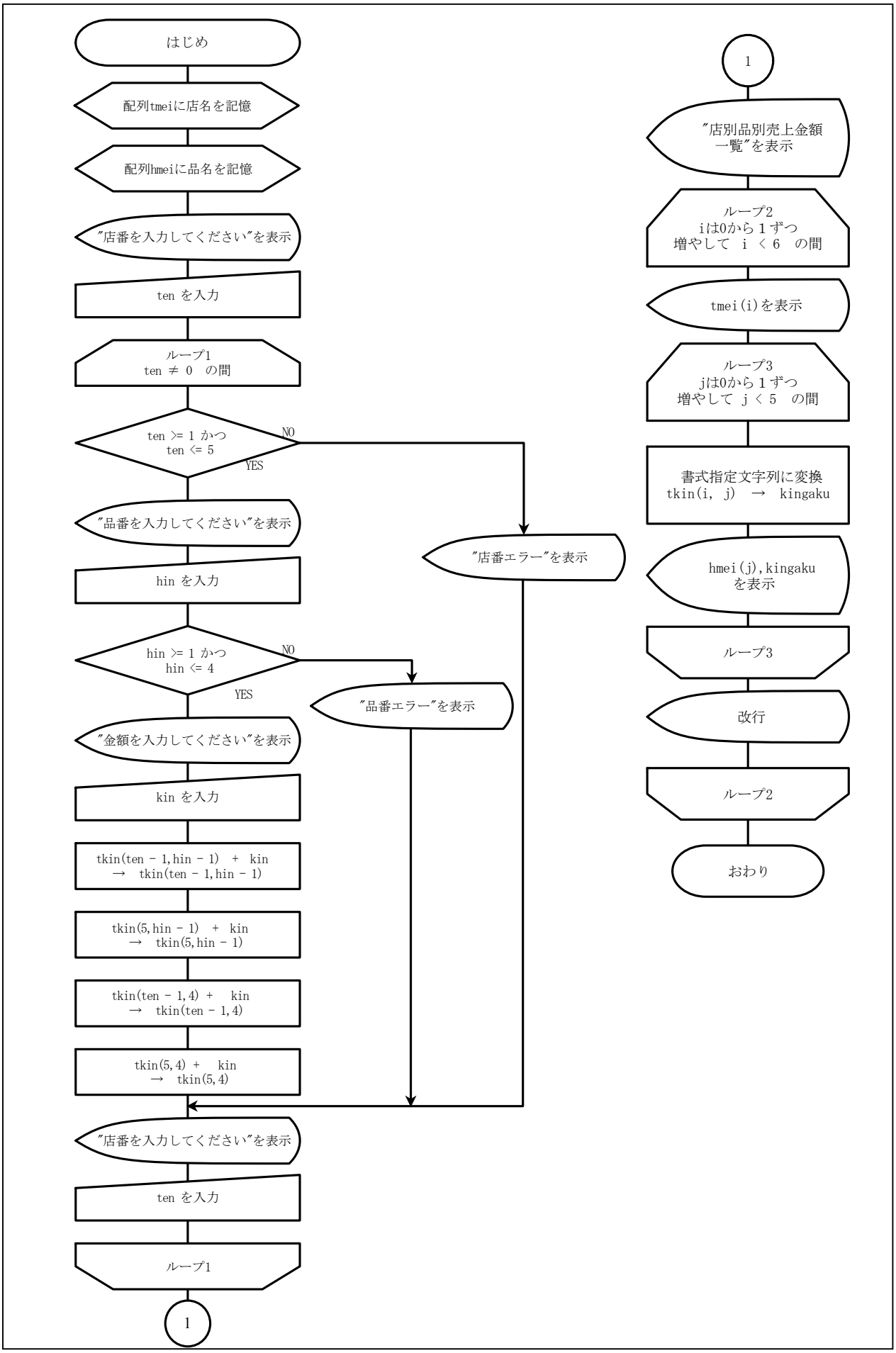
①1 つの main のみでメソッドを使わない場合

②main と 3 つのメソッドで作成した場合

③メソッドに戻り値をつけた場合

【パターン①】1 つの main のみでメソッドを使わない場合

★流れ図 1

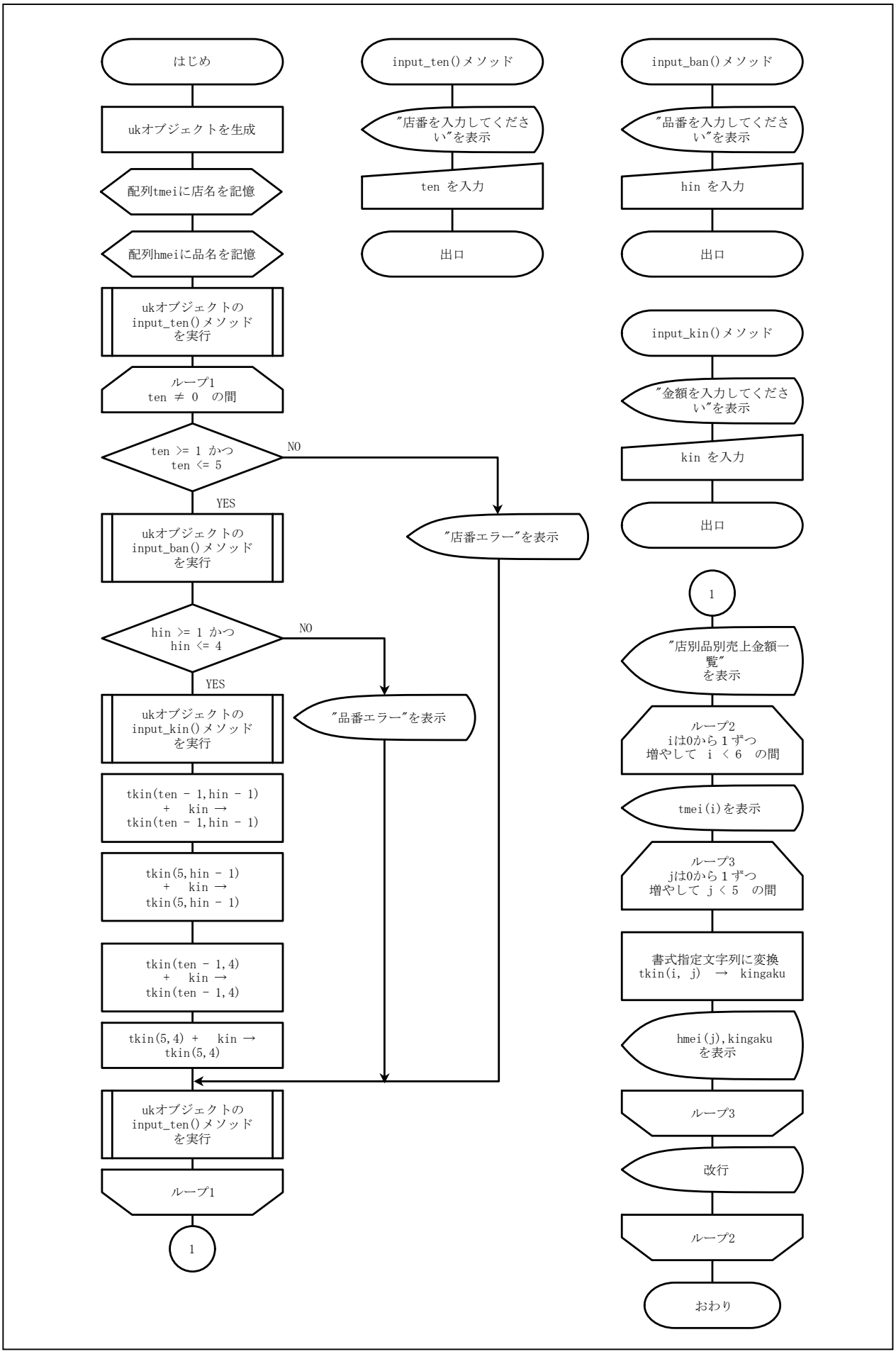


★パターン１のプログラム

```
1  import java.util.*;
2  publicclass UriageKei {
3      publicstaticvoid main(String[] args){
4          int ten,hin, kin;
5          int i, j;
6          int[ ][ ]tkin = newint[6][5];
7          String [ ]tmei = {"神戸支店","姫路支店","赤穂支店","城崎支店","三田支店",
8                            "計      "};
9          String [ ]hmei = {"野球","サッカー","陸上","テニス","計"};
10         Scanner scan = new Scanner(System.in);
11
12         System.out.print("店番を入力してください");
13         ten = scan.nextInt();
14         while(ten != 0){
15             if ( ten >= 1 && ten <= 5 ){
16                 System.out.print("品番を入力してください");
17                 hin = scan.nextInt();
18                 if ( hin >= 1 && hin <= 4 ){
19                     System.out.print("金額を入力してください");
20                     kin = scan.nextInt();
21                     tkin[ten - 1][hin - 1] = tkin[ten - 1][hin - 1] + kin ;
22                     tkin[5][hin - 1] = tkin[5][hin - 1] + kin ;
23                     tkin[ten - 1][4] = tkin[ten - 1][4] + kin ;
24                     tkin[5][4] = tkin[5][4] + kin ;
25                 }
26                 else {
27                 System.out.println("品番エラー");
28                 }
29             }
30             else{
31                 System.out.println("店番エラー");
32             }
33             System.out.print("店番を入力してください");
34             ten = scan.nextInt();
35         }
36         System.out.println("¥t" + "店別品別売上金額一覧");
37         for(i = 0;i < 6; i++){
38             System.out.print( tmei[i] +  "   ");
39             for(j = 0;j < 5; j++){
40                 String kingaku = String.format("%,6d", tkin[i][j]);
41                 // %,6d  6桁で右寄せコンマ付き
42                 System.out.print( hmei[j] + ":" + "¥t"+ kingaku + "¥t" );
43                 }
44             System.out.println();
45         }
46     }
47 }
```

【パターン②】main と３つのメソッドで作成した場合

★流れ図２

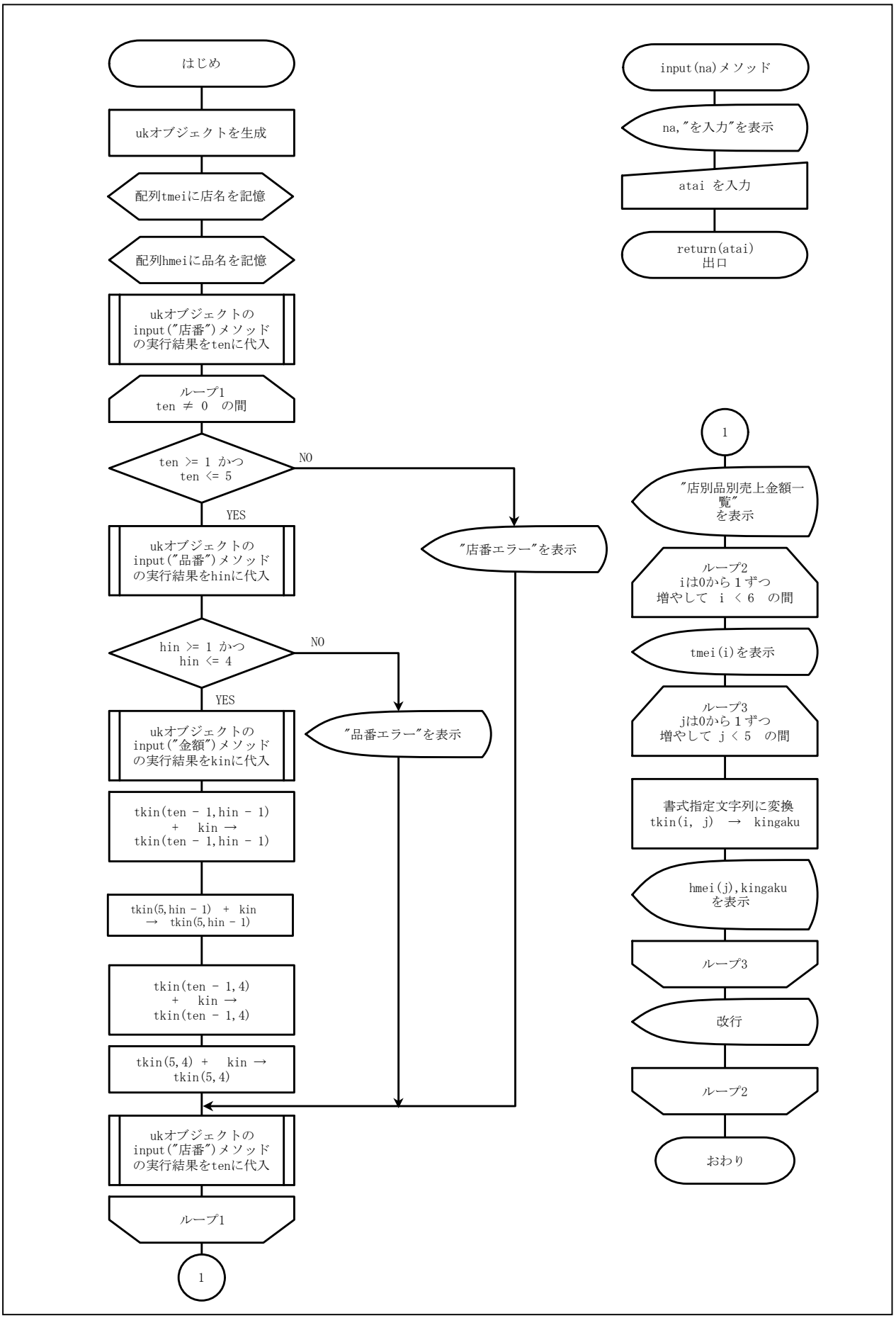


★パターン2のプログラム

```
1  import java.util.Scanner;
2  publicclass UriageKei2 {
3      privatestaticintten, hin, kin;
4      static Scanner scan = new Scanner(System.in);
5      void input_ten(){  //店番入力メソッド
6          System.out.print("店番を入力してください");
7          ten = scan.nextInt();
8      }
9      void input_ban(){  //品番入力メソッド
10         System.out.print("品番を入力してください");
11         hin = scan.nextInt();
12     }
13     void input_kin(){  //金額入力メソッド
14         System.out.print("金額を入力してください");
15         kin = scan.nextInt();
16     }
17     publicstaticvoid main(String[] args) {
18         UriageKei2 uk = new UriageKei2();
19         int i, j;
20         int[ ][ ]tkin = newint[6][5];
21         String [ ]tmei = {"神戸支店","姫路支店","赤穂支店","城崎支店","三田支店",
22                           "計      "};
23         String [ ]hmei = {"野球","サッカー","陸上","テニス","計"};
24         uk.input_ten();  //店番入力メソッド呼び出し
25         while(ten != 0){
26             if ( ten>= 1 &&ten<= 5 ){
27                 uk.input_ban();  //品番入力メソッド呼び出し
28                 if ( hin>= 1 &&hin<= 4 ){
29                     uk.input_kin();  //     金額入力メソッド呼び出し
30                     tkin[ten - 1][hin - 1] += kin ;
31                     tkin[5][hin - 1] += kin ;
32                     tkin[ten - 1][4] += kin ;
33                     tkin[5][4] += kin ;
34                 }
35                 else {
36                 System.out.println("品番エラー");
37                 }
38             }
39             else{
40                 System.out.println("店番エラー");
41             }
42             uk.input_ten();  //店番入力メソッド呼び出し
43         }
44         System.out.println("¥t" + "店別品別売上金額一覧");
45         for(i = 0;i < 6; i++) {
46             System.out.print( tmei[i] + "¥t");
47             for(j = 0;j < 5; j++) {
48                 String kingaku = String.format("%,6d", tkin[i][j]);
49                 System.out.print( hmei[j] + ":" + "¥t"+ kingaku + "¥t" );
50             }
51             System.out.println();
52         }
53     }
54 }
```

【パターン③】メソッドに戻り値をつけた場合

★流れ図３



パターン３のプログラム

```
1  import java.util.Scanner;
2  publicclass UriageKei3 {
3      static Scanner scan = new Scanner(System.in);
4
5      int input(String na){  //入力メソッド
6          System.out.print(na + "を入力");
7          int  atai = scan.nextInt();
8          return  atai;//  入力したataiを返す
9      }
10
11     publicstaticvoid main(String[] args) {
12         UriageKei3 uk = new UriageKei3();
13         int i, j;
14         int ten, hin, kin;
15         int[ ][ ]tkin = newint[6][5];
16         String [ ]tmei = {"神戸支店","姫路支店","赤穂支店","城崎支店","三田支店",
17                           "計      "};
18         String [ ]hmei = {"野球","サッカー","陸上","テニス","計"};
19
20         ten = uk.input("店番");     //入力メソッドを実行、戻り値をtenに代入
21         while(ten != 0){
22             if ( ten >= 1 && ten <= 5 ){
23                 hin = uk.input("品番"); //入力メソッドを実行、戻り値をhinに代入
24                 if ( hin >= 1 && hin <= 4 ){
25                     kin = uk.input("金額"); //入力メソッドを実行、戻り値をkinに代入
26                     tkin[ten - 1][hin - 1] += kin ;
27                     tkin[5][hin - 1] += kin ;
28                     tkin[ten - 1][4] += kin ;
29                     tkin[5][4] += kin ;
30                 }
31                 else {
32                 System.out.println("品番エラー");
33                 }
34             }
35             else{
36                 System.out.println("店番エラー");
37             }
38             ten = uk.input("店番");     //入力メソッドを実行、戻り値をtenに代入
39         }
40         System.out.println("¥t" + "店別品別売上金額一覧");
41         for(i = 0;i <6; i++) {
42             System.out.print( tmei[i] + "¥t");
43             for(j = 0;j < 5; j++) {
44                 String kingaku = String.format("%,6d", tkin[i][j]);
45                 System.out.print( hmei[j] + ":" + "¥t"+ kingaku + "¥t" );
46             }
47             System.out.println();
48         }
49     }
50 }
51
```

(3)配列を利用した処理

①線形探索(数値型)

設問　兵庫商事の売上データをキーボードから入力し、実行結果のように表示するプログラムを作成しなさい。

入力データ

品番	数量
nban	nsu

取り扱う５種類の商品

品番	商品名	単価
11	チーズケーキ	300
13	抹茶ムース	350
15	ザッハトルテ	380
16	ミルフィーユ	420
17	モンブラン	340

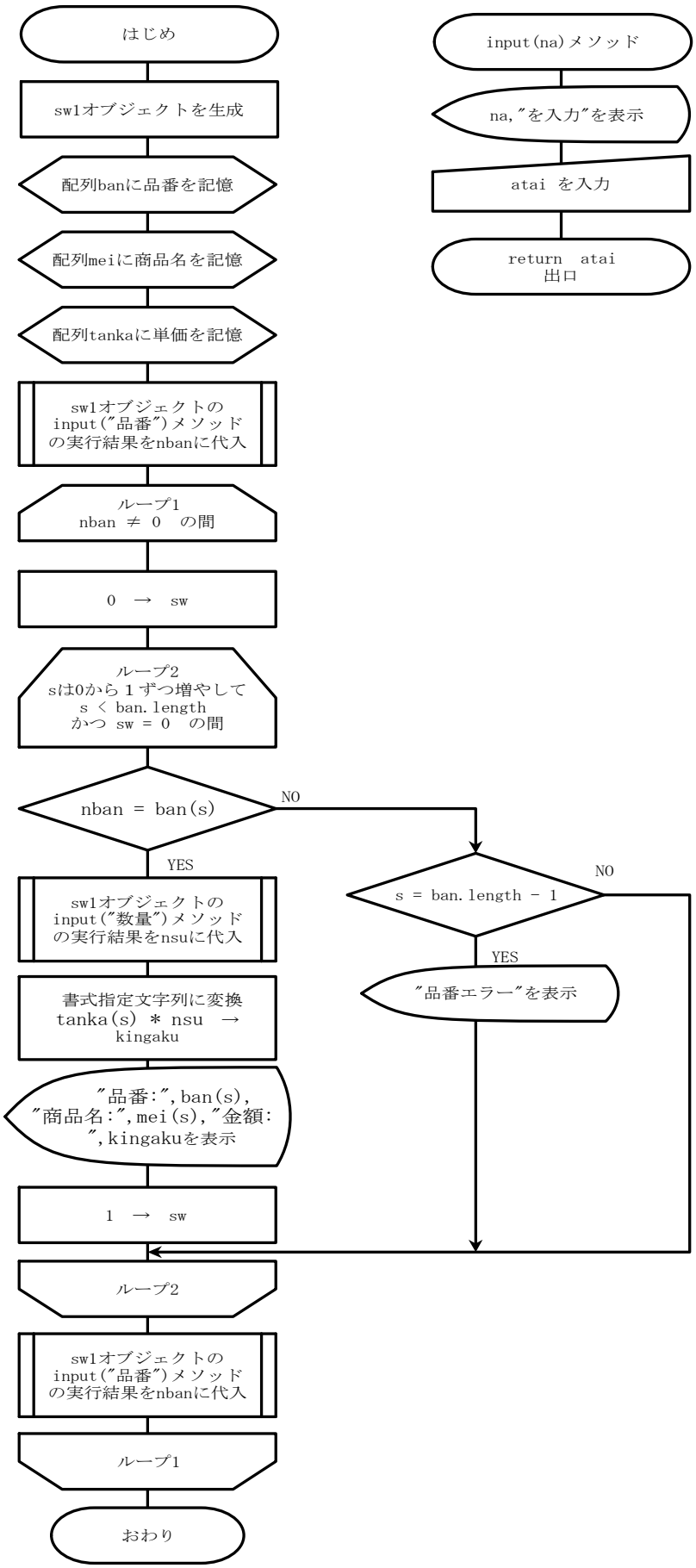
実行結果

```
品番を入力 11
数量を入力 30
  品番:11      商品名:チーズケーキ   金額: 9,000
品番を入力 12
品番エラー
```

処理条件

1. キーボードから品番と数量を入力し、品番によってテーブルから商品名および単価を探求し、金額を求め、表示する。探索によって等しい品番が見つからないときには、品番エラーと表示する。
2. 品番に 0 が入力されたら、プログラムを終了する。

★流れ図


はじめ
sw1オブジェクトを生成
配列banに品番を記憶
配列meiに商品名を記憶
配列tankaに単価を記憶
sw1オブジェクトの
input(″品番″)メソッド
の実行結果をnbanに代入
ループ1
nban ≠ 0 の間
0 → sw
ループ2
sは0から１ずつ増やして
s < ban.length
かつ sw = 0 の間
nban = ban(s)
NO
YES
sw1オブジェクトの
input(″数量″)メソッド
の実行結果をnsuに代入
書式指定文字列に変換
tanka(s) * nsu →
kingaku
″品番:″,ban(s),
″商品名:″,mei(s),″金額:
″,kingakuを表示
1 → sw
s = ban.length - 1
NO
YES
″品番エラー″を表示
ループ2
sw1オブジェクトの
input(″品番″)メソッド
の実行結果をnbanに代入
ループ1
おわり
input(na)メソッド
na,″を入力″を表示
atai を入力
return atai
出口

★プログラム

```
1  import java.util.*;
2  publicclass Sweets1 {
3
4      static Scanner scan = new Scanner(System.in);
5
6      int input(String na){  //入力メソッド
7          System.out.print(na + "を入力");
8          int atai = scan.nextInt();
9          return atai;//  入力したataiを返す
10     }
11
12     publicstaticvoid main(String args[]) {
13
14         Sweets1 sw1 = new  Sweets1();
15
16         int[] ban = {11,13,15,16,17};
17         String[] mei = {"チーズケーキ","抹茶ムース　","ザッハトルテ","ミルフィーユ",
18                         "モンブラン"};
19         int[] tanka = {300,350,380,420,340};
20         int nban, nsu, sw ;
21
22         nban = sw1.input("品番");  //入力メソッドを実行、戻り値をnbanに代入
23         while(nban != 0){
24              // データを線形探索する
25             sw = 0;
26             for(int s = 0; s < ban.length && sw == 0; s++) { // ※1
27                 if(nban == ban[s]) {
28                     nsu = sw1.input("数量");  //入力メソッドを実行、戻り値をnsuに代入
29                     String kingaku = String.format("%,7d", tanka[s] * nsu);
30                     System.out.println(" 品番:"  + ban[s] + "¥t" + " 商品名:"  + mei[s]
31                                         + "¥t" + "金額:" +  kingaku);
32                     sw = 1; // 見つかったらループを終了する
33                 }
34                 else if (s == ban.length -1 ){ // 見つからなかったら
35                     System.out.println("品番エラー");
36                 }
37             }
38             nban = sw1.input("品番");  //入力メソッドを実行し、戻り値をnbanに代入する
39         }
40     }
41 }
```

※1　ban.length 配列変数名.length で配列の要素数(長さ)を取得できる

②線形探索（文字列型）

設問　県名（近畿２府４県）データをキーボードから入力し、実行結果のように表示するプログラムを作成しなさい。

入力データ

県名
mei

近畿２府４県のデータ

府県名	県庁所在地	県花
京都府	京都	しだれ桜
大阪府	大阪	さくらそう
兵庫県	神戸	のじぎく
滋賀県	大津	しゃくなげ
奈良県	奈良	奈良八重桜
和歌山県	和歌山	梅

実行結果

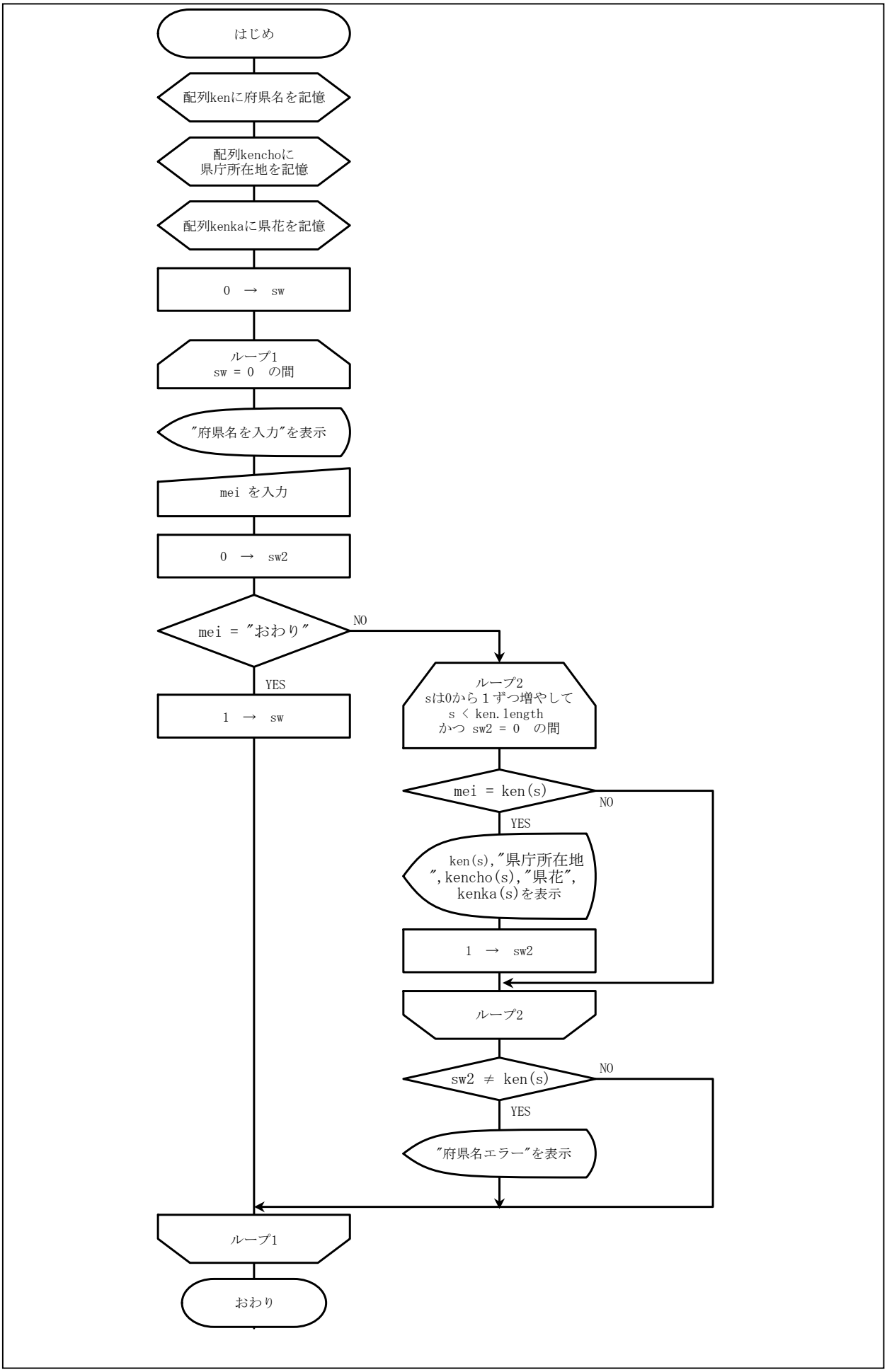
府県名を入力
兵庫県
兵庫県　　県庁所在地　神戸　県花　のじぎく

府県名を入力
石川県
県名エラー

処理条件

1. キーボードから府県名を入力し、府県名によってテーブルから県庁所在地および県花を探索し、表示する。探索によって等しい府県名が見つからないときには、府県名エラーと表示する。
2. 府県名に「おわり」が入力されたら、プログラムを終了する。

★流れ図


はじめ
配列kenに府県名を記憶
配列kenchoに
県庁所在地を記憶
配列kenkaに県花を記憶
0 → sw
ループ1
sw = 0 の間
"府県名を入力"を表示
mei を入力
0 → sw2
mei = "おわり"
NO
YES
1 → sw
ループ2
sは0から１ずつ増やして
s < ken.length
かつ sw2 = 0 の間
mei = ken(s)
NO
YES
ken(s),"県庁所在地
",kencho(s),"県花",
kenka(s)を表示
1 → sw2
ループ2
sw2 ≠ ken(s)
NO
YES
"府県名エラー"を表示
ループ1
おわり

★プログラム

```
1  import java.util.*;
2  public class Kenmei {
3      public static void main(String args[]) {
4          String[] ken = {"京都府","大阪府","兵庫県","滋賀県","奈良県","和歌山県"};
5          String[] kencho = {"京都","大阪","神戸","大津","奈良","和歌山"};
6          String[] kenka = {"しだれ桜","さくらそう","のじぎく","しゃくなげ",
7  "奈良八重桜","梅"};
8          String mei;
9          int sw = 0;
10         int sw2;
11         Scanner scan = new Scanner(System.in);
12         while(sw == 0){
13             System.out.println("府県名を入力");
14             mei = scan.next();
15             sw2 = 0;
16             if(mei.equals("おわり")) {
17                 sw = 1;
18                 }
19             else {
20                 for(int s = 0;s < ken.length && sw2 == 0 ;s++) {
21                     if(mei.equals(ken[s])) {
22                         System.out.println (ken[s] + "  県庁所在地   "+ kencho[s]
23  + "  県花  " +  kenka[s]);
24                         sw2= 1;
25                         }
26                 }
27                 if (sw2 != 1) System.out.println("府県名エラー");
28             }
29         }
30     }
31 }
```

③並べ替え

設問１　兵庫商事の売上データをキーボードから入力し、実行結果のように表示するプログラムを作成しなさい。

入力データ

販売員コード hco	売上高 uri

実行結果

```
販売員コードを入力
1
売上高を入力
30
  ⋮
  販売員コード　売　上　高
        5          8,000
        2          6,000
      ⋮
        3          2,000
```

処理条件

1. キーボードから販売員コードと売上高を入力し、配列 tcode および turi に記憶する。

   配列 tcode

(0)	(1)	(2)	(3)	(4)

   配列 turi

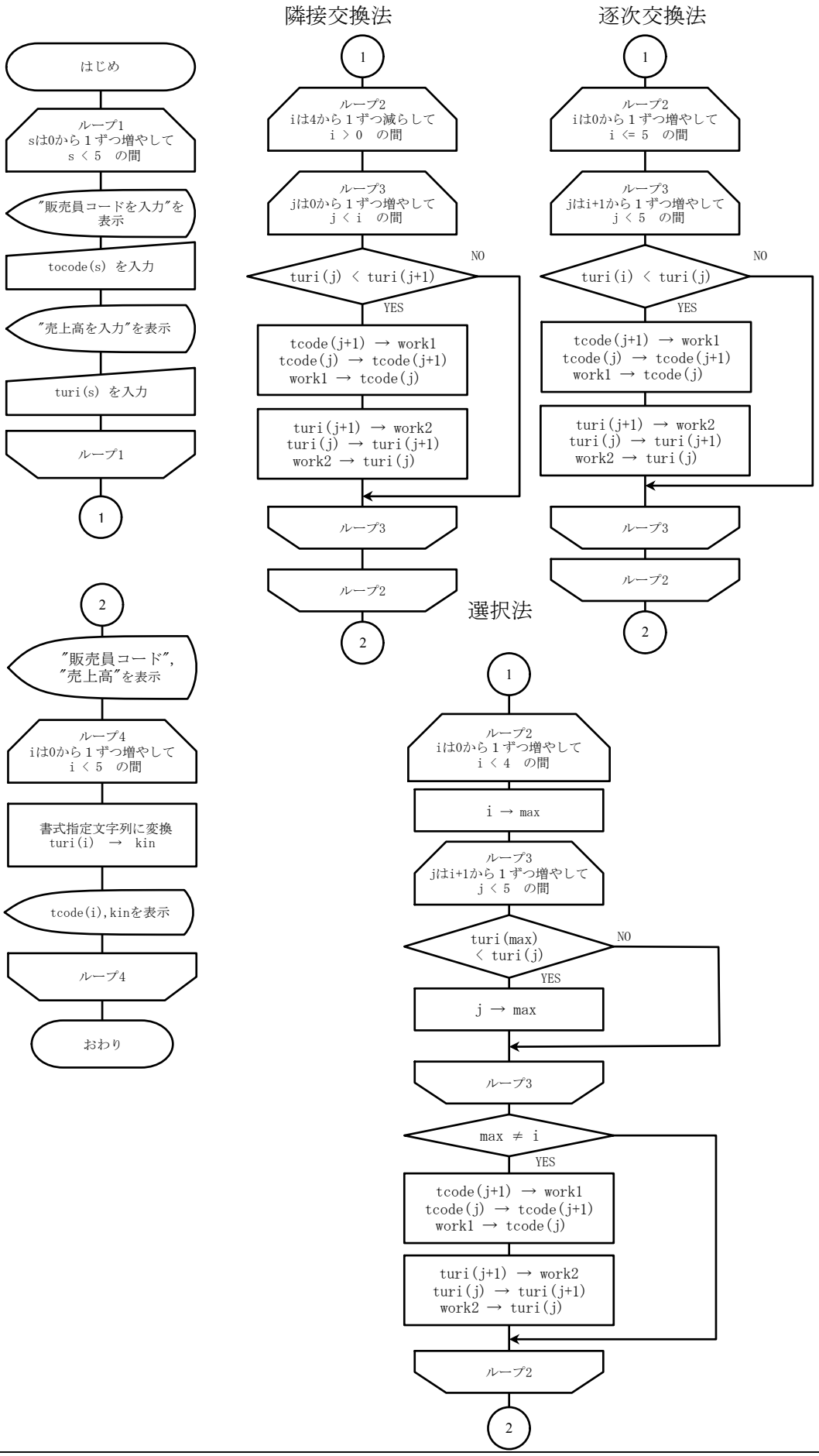
(0)	(1)	(2)	(3)	(4)

2. データ件数は 5 である。
3. 販売員コードに 0 が入力されたら入力を終了し、売上高の降順に並べ替えて表示する。

※　並べ替え方法について次の３方法を示している。

- ・隣接交換法　　（プログラム１）
- ・逐次交換法　　(プログラム 2)
- ・選択法　　(プログラム 3)

★流れ図


はじめ
ループ1
sは0から１ずつ増やして
s ＜ 5　の間
"販売員コードを入力"を
表示
tocode(s) を入力
"売上高を入力"を表示
turi(s) を入力
ループ1
1
2
"販売員コード",
"売上高"を表示
ループ4
iは0から１ずつ増やして
i ＜ 5　の間
書式指定文字列に変換
turi(i) → kin
tcode(i),kinを表示
ループ4
おわり
隣接交換法
1
ループ2
iは4から１ずつ減らして
i ＞ 0　の間
ループ3
jは0から１ずつ増やして
j ＜ i　の間
turi(j) ＜ turi(j+1)
NO
YES
tcode(j+1) → work1
tcode(j) → tcode(j+1)
work1 → tcode(j)
turi(j+1) → work2
turi(j) → turi(j+1)
work2 → turi(j)
ループ3
ループ2
2
逐次交換法
1
ループ2
iは0から１ずつ増やして
i <= 5　の間
ループ3
jはi+1から１ずつ増やして
j ＜ 5　の間
turi(i) ＜ turi(j)
NO
YES
tcode(j+1) → work1
tcode(j) → tcode(j+1)
work1 → tcode(j)
turi(j+1) → work2
turi(j) → turi(j+1)
work2 → turi(j)
ループ3
ループ2
2
選択法
1
ループ2
iは0から１ずつ増やして
i ＜ 4　の間
i → max
ループ3
jはi+1から１ずつ増やして
j ＜ 5　の間
turi(max)
＜ turi(j)
NO
YES
j → max
ループ3
max ≠ i
YES
tcode(j+1) → work1
tcode(j) → tcode(j+1)
work1 → tcode(j)
turi(j+1) → work2
turi(j) → turi(j+1)
work2 → turi(j)
ループ2
2

プログラム 1（隣接交換法）

```
 1  import java.util.*;
 2  public class Sort1 {
 3      public static void main(String args [])   {
 4          int [] tcode =new int[5];
 5          int [] turi =new int[5];
 6          int work1, work2;
 7          int s, i, j;
 8          Scanner scan = new Scanner(System.in);
 9
10          // データを記憶する
11          for(s = 0; s < 5 ; s++) {
12              System.out.println("販売員コードを入力");
13              tcode[s] = scan.nextInt();
14              System.out.println("売上高を入力");
15              turi[s] = scan.nextInt();
16          }
17
18          //  並べ替え 隣接交換
19          for(i = 4; i > 0; i--) {
20              for(j = 0; j < i ; j++ ) {
21                  if(turi[j] < turi[j + 1]) {
22                      work1  = tcode[j + 1];
23                      tcode[j + 1] = tcode[j];
24                      tcode[j] = work1;
25
26                      work2  = turi[j + 1];
27                      turi[j + 1] = turi[j];
28                      turi[j] = work2;
29                  }
30              }
31          }
32          System.out.println("販売員コード" + "¥t" + "売　上　高");
33          for(i = 0;i < 5;i++){
34              String kin = String.format("%,7d",  turi[i]);
35              System.out.println("¥t" + tcode[i] + "¥t" +  kin);
36          }
37      }
43  }
44
```

プログラム２（逐次交換法）

```
1  import java.util.*;
2  public class Sort2 {
3      public static void main(String args [])    {
4          int [] tcode =new int[5];
5          int [] turi =new int[5];
6          int work1, work2;
7          int s, i, j;
8          Scanner scan = new Scanner(System.in);
9
10         // データを記憶する
11         for(s = 0; s < 5 ; s++) {
12             System.out.println("販売員コードを入力");
13             tcode[s] = scan.nextInt();
14             System.out.println("売上高を入力");
15             turi[s] = scan.nextInt();
16         }
17                 //  並べ替え 逐次交換
18         for(i = 0; i <= 5; i++)  {
19             for(j = i + 1 ; j < 5 ; j++ )    {
20                 if(turi[i] < turi[j]) {
21                     work1  = tcode[i];
22                     tcode[i] = tcode[j];
23                     tcode[j] = work1;
24
25                     work2  = turi[i];
26                     turi[i] = turi[j];
27                     turi[j] = work2;
28                 }
29             }
30         }
31         System.out.println("販売員コード" + "¥t" + "売　上　高");
32         for(i = 0;i < 5;i++){
33             String kin = String.format("%,7d",  turi[i]);
34             System.out.println("¥t" + tcode[i] + "¥t" +  kin);
35          }
36     }
37 }
```

プログラム３（選択法）

```
 1  import java.util.*;
 2  public class Sort3 {
 3      public static void main(String args [])   {
 4          int [] tcode =new int[5];
 5          int [] turi =new int[5];
 6          int work1, work2;
 7          int s, i, j, max;
 8          Scanner scan = new Scanner(System.in);
 9
10          // データを記憶する
11          for(s = 0; s < 5 ; s++) {
12              System.out.println("販売員コードを入力");
13              tcode[s] = scan.nextInt();
14              System.out.println("売上高を入力");
15              turi[s] = scan.nextInt();
16          }
17
18          //  並べ替え 選択法
19          for(i = 0; i < 4; i++) {
20              max = i;
21              for(j = i + 1 ; j < 5 ; j++ ) {
22                  if(turi[max] < turi[j])  {
23                      max = j;
24                  }
25              }
26              if(max != i) {
27                  work1  = tcode[i];
28                  tcode[i] = tcode[max];
29                  tcode[max] = work1;
30
31                  work2  = turi[i];
32                  turi[i] = turi[max];
33                  turi[max] = work2;
34              }
35          }
36          System.out.println("販売員コード" + "¥t" + "売　上　高");
37          for(i = 0;i < 5;i++){
38              String kin = String.format("%,7d",  turi[i]);
39              System.out.println("¥t" + tcode[i] + "¥t" +  kin);
40           }
41      }
42  }
```

④順位づけ

設問　兵庫商事の売上データをキーボードから入力し、実行結果のように表示するプログラムを作成しなさい。

入力データ

販売員コード hco	売上高 uri

実行結果

```
販売員コードを入力
102
売上高を入力
500000
 ⋮
販売員コード　売上高　　順位
　　102　　500,000　　4
　　115　　300,000　　5
　　⋮
　　210　　900,000　　1
```

処理条件

1. キーボードから販売員コードと売上高を入力し、配列 tcode および turi に記憶する。

   配列 tcode

(0)	(1)	(2)	(3)	(4)

   配列 turi

(0)	(1)	(2)	(3)	(4)

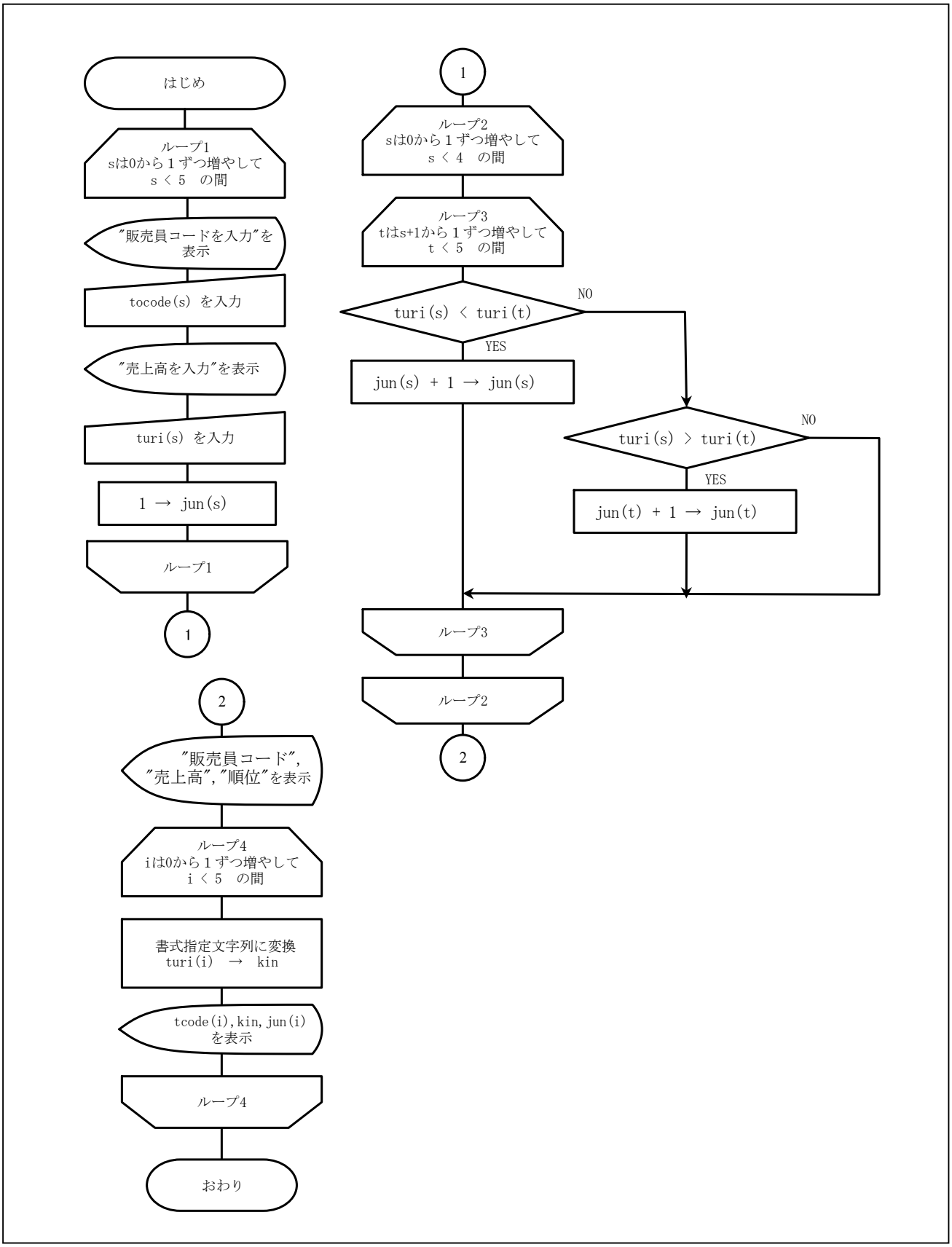
2. 配列 jun を利用して、売上高の降順に順位を付ける

   配列 jun

(0)	(1)	(2)	(3)	(4)

3. データ件数は 5 である。
4. 販売員コードに 0 が入力されたら入力を終了し、販売員コード・売上高・順位を表示する。

★流れ図


はじめ
ループ1
sは0から１ずつ増やして
s < 5 の間
"販売員コードを入力"を表示
tocode(s) を入力
"売上高を入力"を表示
turi(s) を入力
1 → jun(s)
ループ1
1
1
ループ2
sは0から１ずつ増やして
s < 4 の間
ループ3
tはs+1から１ずつ増やして
t < 5 の間
turi(s) < turi(t)
NO
YES
jun(s) + 1 → jun(s)
turi(s) > turi(t)
NO
YES
jun(t) + 1 → jun(t)
ループ3
ループ2
2
2
"販売員コード",
"売上高","順位"を表示
ループ4
iは0から１ずつ増やして
i < 5 の間
書式指定文字列に変換
turi(i) → kin
tcode(i),kin,jun(i)
を表示
ループ4
おわり

★プログラム

```
1   import java.util.Scanner;
2   public class Juni {
3       public static void main(String args [])   {
4           int [] tcode = new int[5];
5           int [] turi = new int[5];
6           int [] jun = new int[5];
7           int s, t, i;
8           Scanner scan = new Scanner(System.in);
9
10          // データを記憶する
11          for(s = 0; s < 5 ; s++) {
12              System.out.println("販売員コードを入力");
13              tcode[s] = scan.nextInt();
14              System.out.println("売上高を入力");
15              turi[s] = scan.nextInt();
16              jun[s] = 1;
17          }
18          //     順位づけをする
19          for(s = 0; s < 4; s++)
20          {
21              for(t = s + 1; t < 5; t++)
22              {
23                  if(turi[s] < turi[t])
24                  {
25                      jun[s]= jun[s] + 1;
26                  }
27                  else if(turi[s] > turi[t])
28                      {
29                          jun[t]= jun[t] + 1;
30                      }
31              }
32          }
33
34          System.out.println("販売員コード" + "¥t" + "売上高"+ "¥t" + "順位");
35          for(i = 0;i < 5;i++){
36              String kin = String.format("%,6d",  turi[i]);
37              System.out.println("¥t" + tcode[i] + "¥t" +  kin+ "¥t" + jun[i]);
38          }
39      }
40  }
```

⑤二分探索（数値型）

設問　　兵庫商事の売上データをキーボードから入力し、実行結果のように表示するプログラムを作成しなさい。

入力データ

品番	数量
nban	nsu

取り扱う５種類の商品

品番	商品名	単価
11	チーズケーキ	300
13	抹茶ムース	350
15	ザッハトルテ	380
16	ミルフィーユ	420
17	モンブラン	340

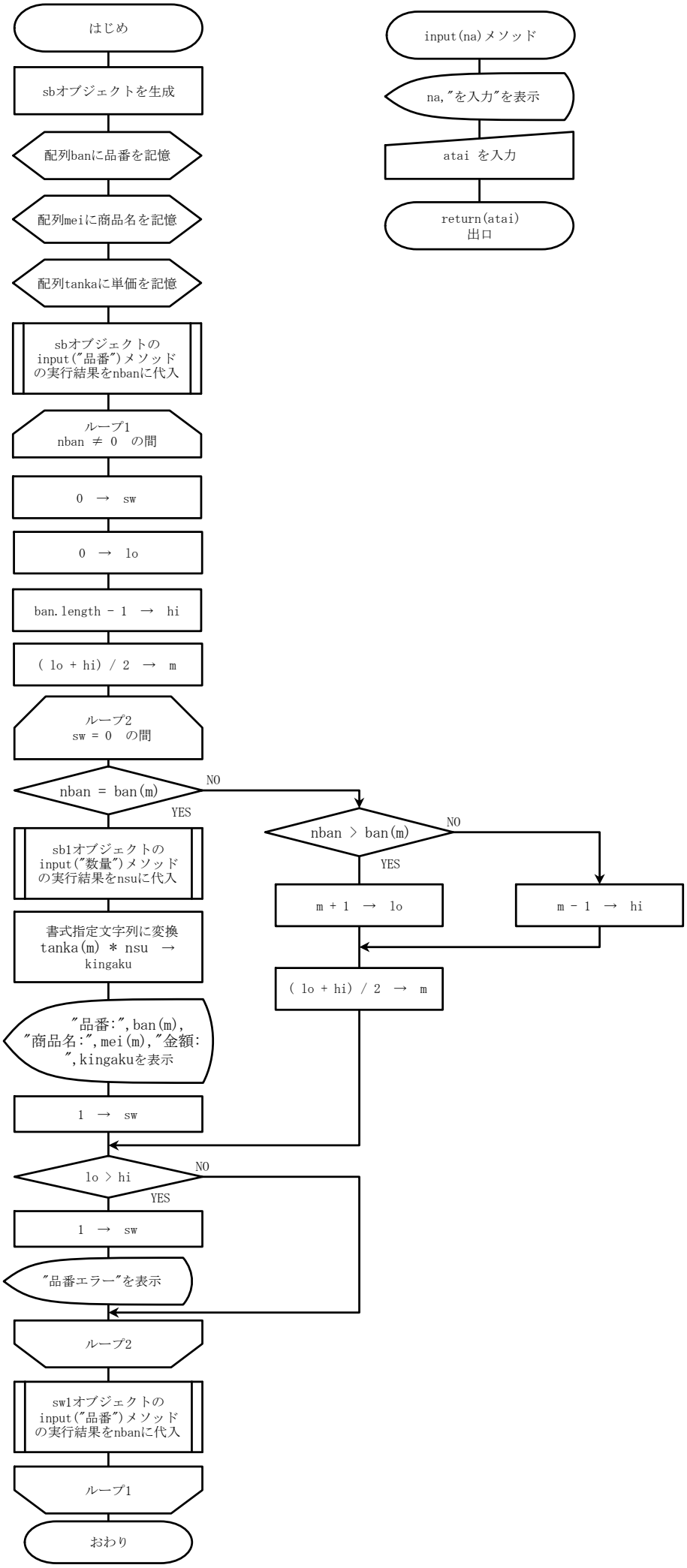
実行結果

```
品番を入力
11
数量を入力
30
  品番 11      商品名   チーズケーキ   金額   9,000
品番を入力
12
品番エラー
```

処理条件

1. キーボードから品番と数量を入力し、品番によってテーブルから商品名および単価を検索し、金額を求め、表示する。検索によって等しい品番が見つからないときには、品番エラーと表示する。
2. 品番に 0 が入力されたら、プログラムを終了する。

★流れ図


はじめ
sbオブジェクトを生成
配列banに品番を記憶
配列meiに商品名を記憶
配列tankaに単価を記憶
sbオブジェクトの
input("品番")メソッド
の実行結果をnbanに代入
ループ1
nban ≠ 0 の間
0 → sw
0 → lo
ban.length - 1 → hi
(lo + hi) / 2 → m
ループ2
sw = 0 の間
nban = ban(m)
NO
YES
sb1オブジェクトの
input("数量")メソッド
の実行結果をnsuに代入
書式指定文字列に変換
tanka(m) * nsu →
kingaku
"品番:",ban(m),
"商品名:",mei(m),"金額:
",kingakuを表示
1 → sw
lo > hi
NO
YES
1 → sw
"品番エラー"を表示
ループ2
sw1オブジェクトの
input("品番")メソッド
の実行結果をnbanに代入
ループ1
おわり
nban > ban(m)
NO
YES
m + 1 → lo
m - 1 → hi
(lo + hi) / 2 → m
input(na)メソッド
na,"を入力"を表示
atai を入力
return(atai)
出口

★プログラム

```
1   import java.util.*;
2   public class Sweets_BinarySearch {
3
4       static Scanner scan = new Scanner(System.in);
5       int input(String na){  //入力メソッド
6           System.out.print(na + "を入力");
7           int atai = scan.nextInt();
8           return atai;//  入力した atai を返す
9       }
10
11      public static void main(String args[]) {
12          Sweets_BinarySearch sb = new  Sweets_BinarySearch();
13          int[] ban = {11,13,15,16,17};
14          String[] mei = {"チーズケーキ","抹茶ムース  ","ザッハトルテ",
15  "ミルフィーユ","モンブラン"};
16          int[] tanka = {300,350,380,420,340};
17          int nban, nsu, sw;
18          int hi, lo, m;
19
20          nban = sb.input("品番");  //入力メソッドを実行し、戻り値を nban に代入する
21          while(nban != 0){
22              sw = 0;
23              lo = 0;
24              hi = ban.length - 1;
25              m = (lo + hi)/ 2; //中央値を求める
26              while(sw == 0){
27                  if (nban == ban[m]){
28                      nsu = sb.input("数量");
29                      //入力メソッドを実行し、戻り値を nsu に代入する
30                      String kingaku = String.format("%,7d", tanka[m] * nsu);
31                      System.out.println(" 品番:"  + ban[m] + "¥t" + " 商品名:"  + mei[m]
32  + "¥t" + "金額:" +  kingaku);
33                      sw  = 1; // ループを終了する
34                  }
35                  else if(nban > ban[m]) {
36                      lo = m + 1;
37                      }
38                      else {
39                          hi = m - 1;
40                      }
41                  m = (lo + hi)/ 2;
42                  if(lo > hi) {
43                      System.out.println("品番エラー");
44                      sw = 1;
45                  }
46              }
47              nban = sb.input("品番");  //入力メソッドを実行し、戻り値を nban に代入する
48          }
49      }
50  }
```

第4章　総合演習

（1）「オブジェクトごとにメソッドを作成する」

点数を入力し、判定結果を画面に表示するプログラムを作成しなさい。

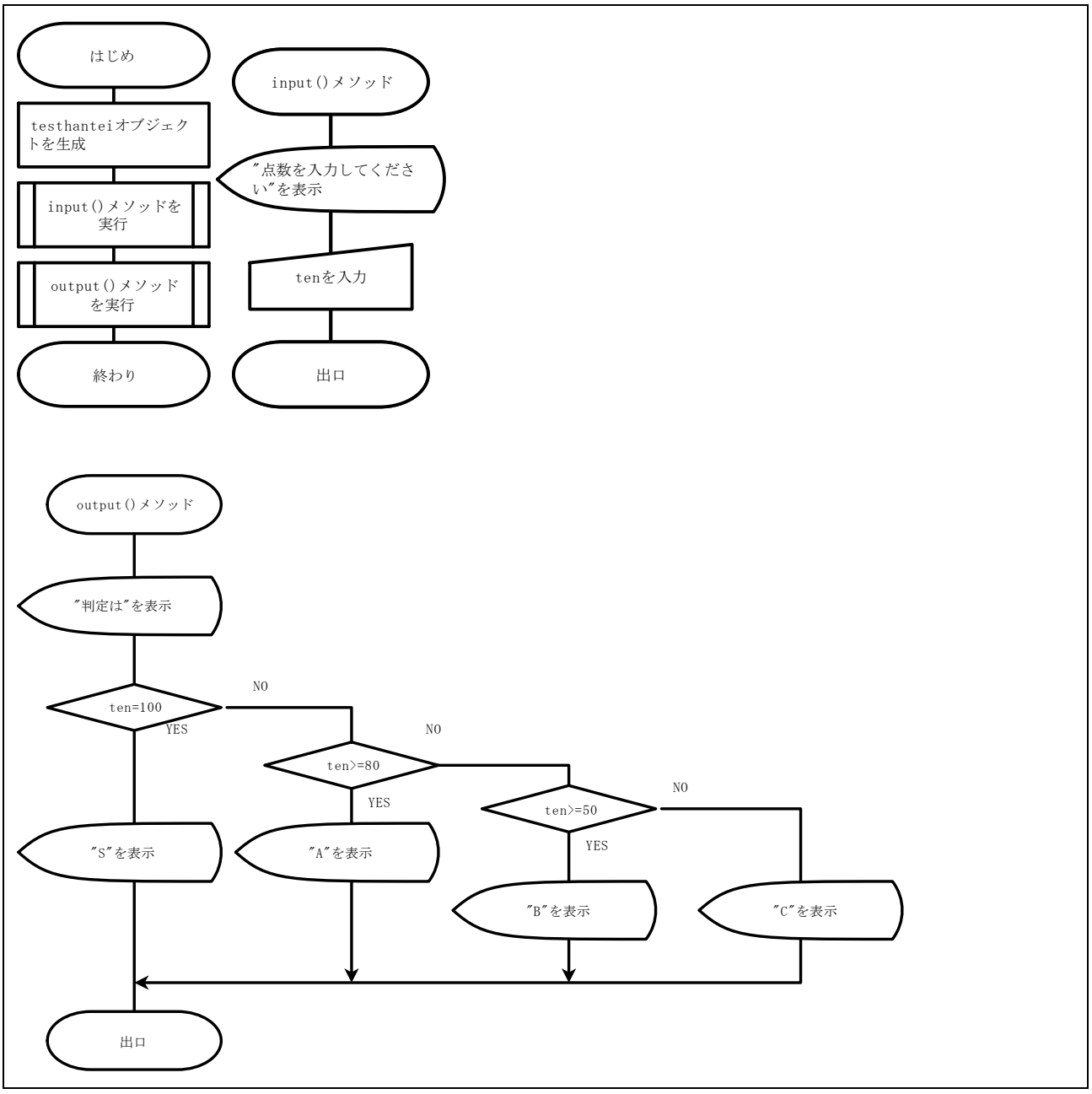
入力データ

得点 ten

実行結果

点数を入力してください 70 判定はB

流れ図

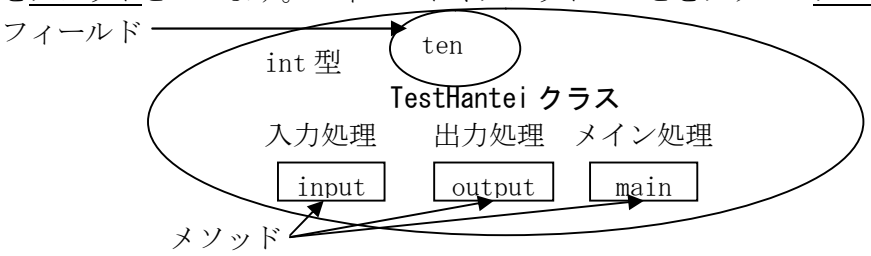


プログラム例

```
 1 import java.util.*;
 2 public class TestHantei {
 3         int ten;
 4         public void input(){
 5                 Scanner scan=new Scanner(System.in);
 6                 System.out.println("点数を入力してください");
 7                 ten=scan.nextInt();
 8         }
 9         public void output(){
10                 System.out.print("判定は");
11                 if(ten == 100){
12                         System.out.println("S");
13                 }else if(ten >= 80){
14                         System.out.println("A");
15                 }else if(ten >= 50){
16                         System.out.println("B");
17                 }else{
18                         System.out.println("C");
19                 }
20         }
21         public static void main(String[] args) {
22                 TestHanteitesthantei=new TestHantei();
23                 testhantei.input();
24                 testhantei.output();
25         }
26 }
```

クラス

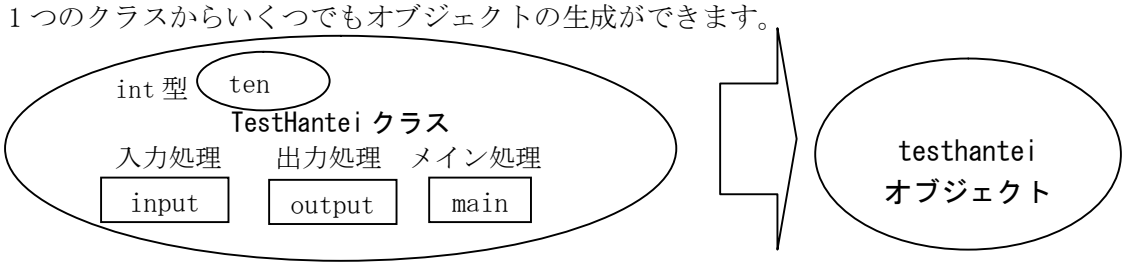
データと処理をひとまとめにしたものをクラスといいます。データのことをフィールド、処理のことをメソッドといいます。フィールドやメソッドのことをクラスのメンバといいます。



オブジェクト

クラスは変数に対する型のようなもので、それ自体に値を格納することはできない。そこでクラスをもとに、値を格納できる変数のようなものを作ります。これをオブジェクトといいます。

1 つのクラスからいくつでもオブジェクトの生成ができます。



（2）「オブジェクトの作成」

国語と英語の点数を入力し、判定結果を画面に表示するプログラムを作成しなさい。

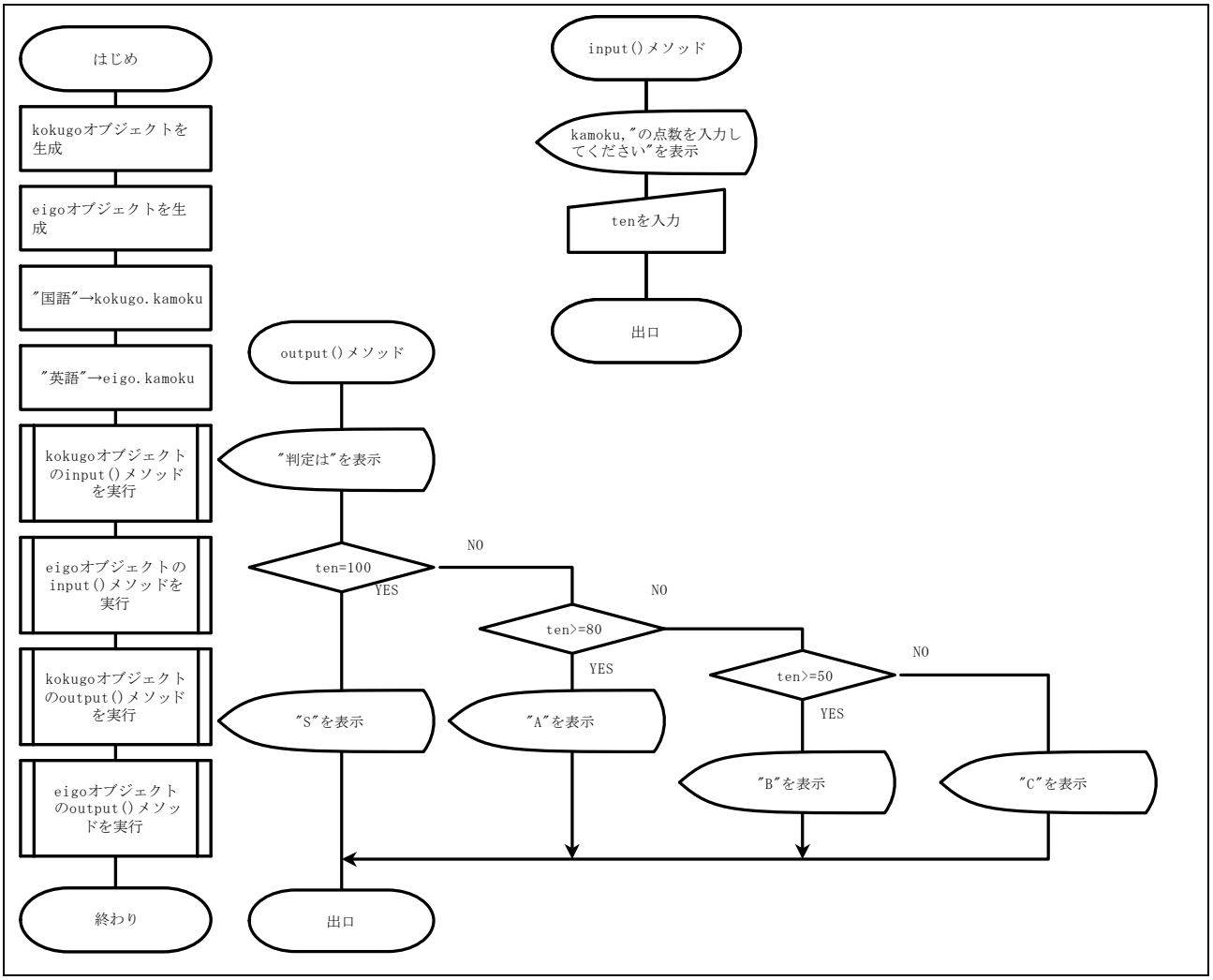
入力データ

得点 ten

実行結果

```
国語の点数を入力してください
70
英語の点数を入力してください
80
国語の判定はＢ
英語の判定はＡ
```

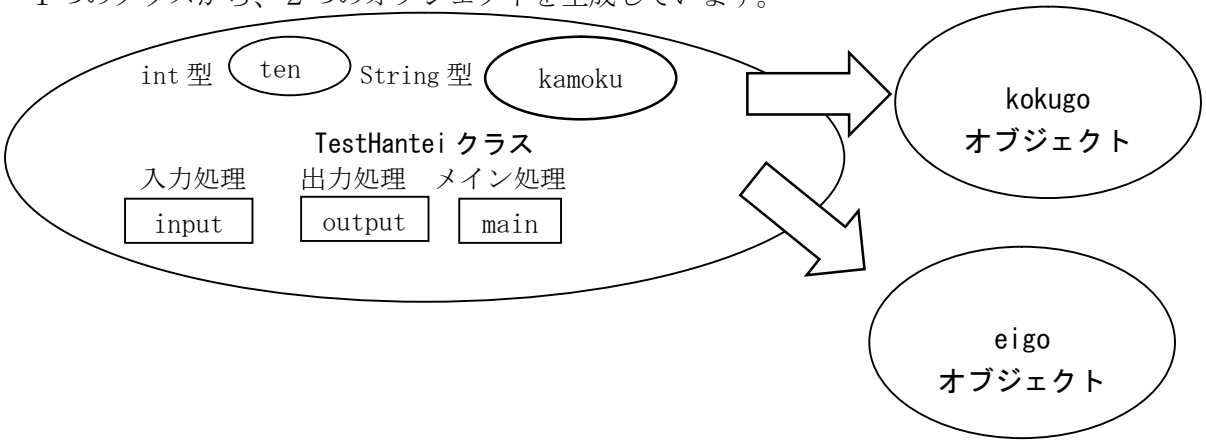
流れ図



プログラム例

```
1  import java.util.*;
2  public class TestHantei {
3          int ten;
4          String kamoku;
5          public void input(){
6                  Scanner scan = new Scanner(System.in);
7                  System.out.println(kamoku + "の点数を入力してください");
8                  ten = scan.nextInt();
9          }
10         public void output(){
11                 System.out.print(kamoku + "の判定は");
12                 if(ten == 100){
13                         System.out.println("S");
14                 }else if(ten >= 80){
15                         System.out.println("A");
16                 }else if(ten >= 50){
17                         System.out.println("B");
18                 }else{
19                         System.out.println("C");
20                 }
21         }
22         public static void main(String[] args) {
23                 TestHanteikokugo = new TestHantei();
24                 TestHanteieigo = new TestHantei();
25                 kokugo.kamoku = "国語";
26                 eigo.kamoku = "英語";
27                 kokugo.input();
28                 eigo.input();
29                 kokugo.output();
30                 eigo.output();
31         }
32 }
```

１つのクラスから、２つのオブジェクトを生成しています。



(3)「最大値を求める」

身長と体重を複数入力し、それぞれの最大値を求めるプログラムを作成しなさい。

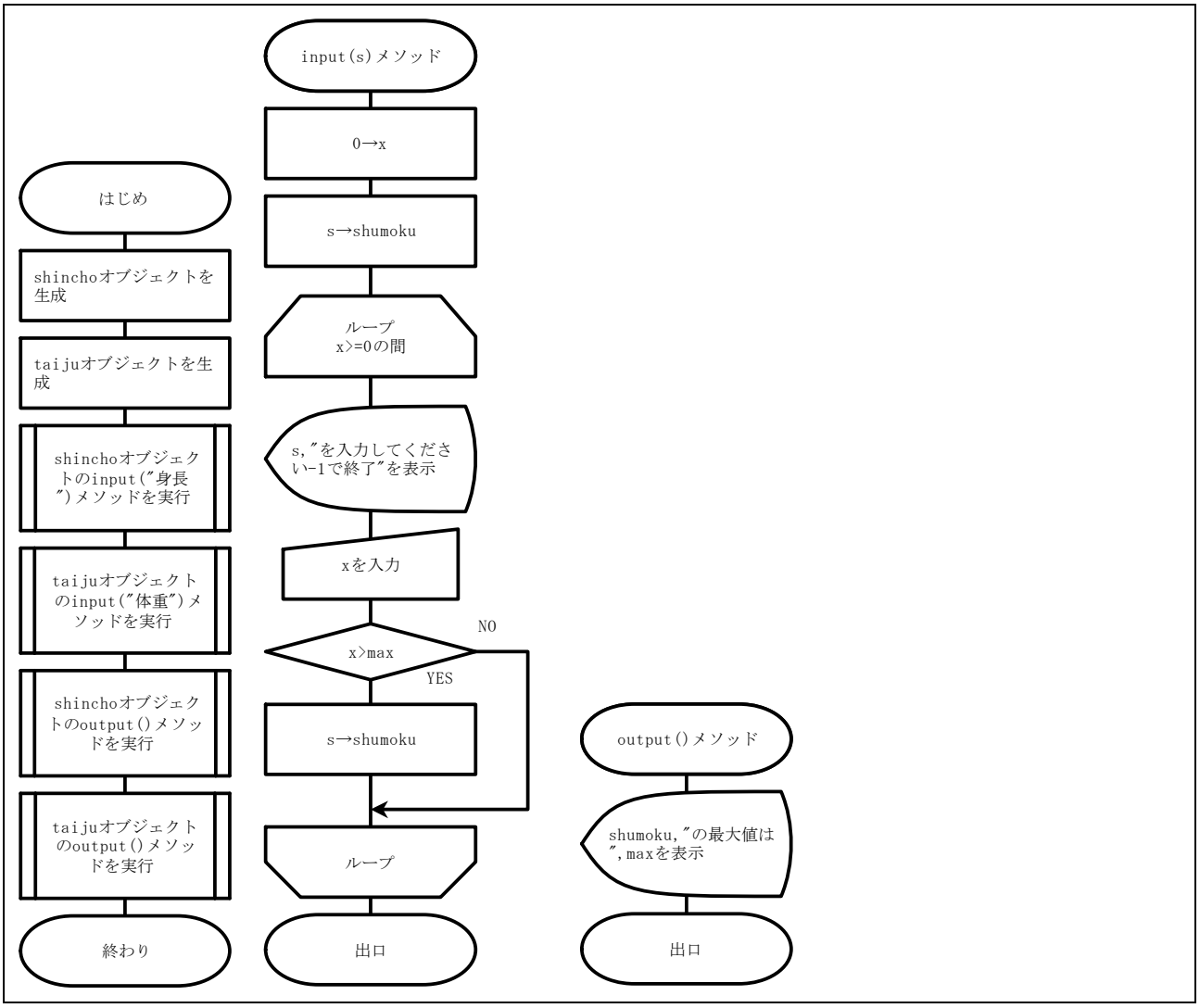
入力データ

身長（体重） x

実行結果

```
身長を入力してください-1 で終了
170
180
160
-1
体重を入力してください-1 で終了
80
75
65
-1
身長の最大値は 180
体重の最大値は 80
```

流れ図



プログラム例

```
1  import java.util.Scanner;
2  public class ShintaiKeisoku {
3          int max = 0;
4          String shumoku;
5          public void input(String s){
6                  Scanner scan = new Scanner(System.in);
7                  int x = 0;
8                  shumoku = s;
9                  System.out.println(s + "を入力してください-1 で終了");
10                 while(x >= 0){
11                         x = scan.nextInt();
12                         if(x > max){
13                                 max = x;
14                         }
15                 }
16         }
17         public void output(){
18                 System.out.println(shumoku + "の最大値は" + max);
19         }
20         public static void main(String[] args) {
21                 ShintaiKeisokushincho = new ShintaiKeisoku();
22                 ShintaiKeisokutaiju = new ShintaiKeisoku();
23                 shincho.input("身長");
24                 taiju.input("体重");
25                 shincho.output();
26                 taiju.output();
27         }
28 }
```

(4)「メソッドからの戻り値の利用」

数値(整数)を入力し、その数値の階乗を表示するプログラムを作成しなさい。ただし、別ファイルにて Kaijo クラスを用意して階乗を求める kaijo メソッドを作成し、それを利用すること。

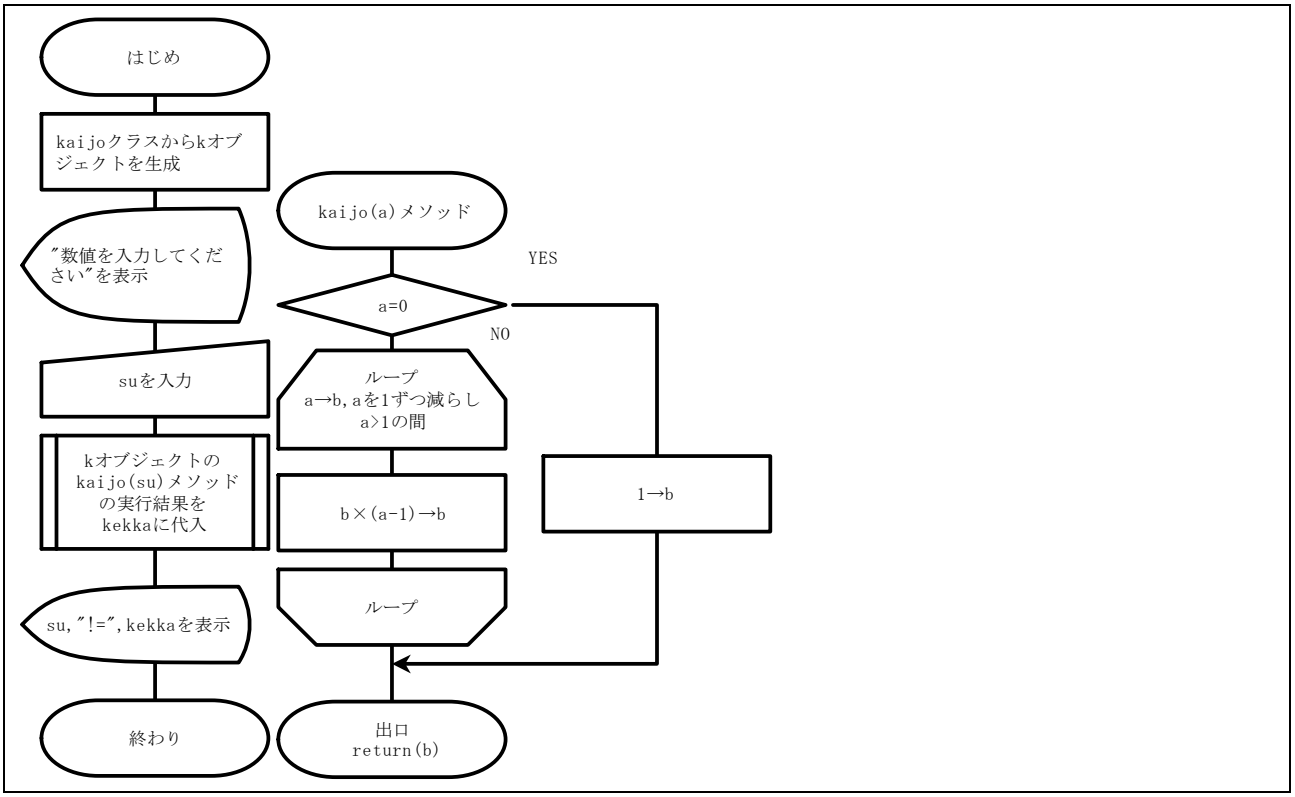
入力データ

数値
su

実行結果

```
数値を入力してください
5
5!=120
```

流れ図



プログラム例：Factorial.java

```
1  import java.util.*;
2  public class Factorial {
3          public static void main(String[] args) {
4                  intsu,kekka;
5                  Scanner scan =new Scanner(System.in);
6                  Kaijo k = new Kaijo();
7                  System.out.println("数値を入力してください");
8                  su = scan.nextInt();
9                  kekka=k.kaijo(su);
10                 System.out.println(su + "!=" + kekka);
11         }
12 }
```

プログラム例：Kaijo.java

```
1  public class Kaijo {
2          public intkaijo(int a){
3                  int b;
4                  if(a == 0){
5                          b = 1;
6                  }else{
7                          for(b = a ; a > 1 ; a --){
8                                  b = b * (a - 1);
9                          }
10                 }
11                 return b;
12         }
13 }
```

第5章　Javaアプレットの作成

Ｊａｖａアプレットは、Ｊａｖａプログラムの実行形態のひとつで、ネットワークを通じてＷｅｂブラウザ上に読み込まれて実行されるプログラムで、チャットやＣＧアニメーション、教育機関による学習システムなどで利用されています。

まずは、「Hello World」と表示するだけのアプレットを作ってみましょう

プログラム例：HelloApplet.java

```
1  import java.applet.Applet;
2  import java.awt.Graphics;
3  public class HelloApplet extends Applet {
4          public void paint(Graphics g){
5                  g.drawString("Hello World!",0,12);
6          }
7  }
```

３行目のクラス名のあとに「extends Applet」と記述されています。これは、すでにある Applet クラスを利用して新しいクラスを作成するときに記述します。作成されるクラスはもとになったクラスの性質を引き継ぎ、すべてのフィールドとメソッドが新しいクラスでも利用可能になっています。

このように既に定義されているクラスをもとに、拡張や変更を加えた新しいクラスを定義することを「継承」といいます。また、Applet クラスを利用するために、１行目で applet クラスをインポートしています。

４行目～６行目は paint メソッドといい、アプレットに表示する文字列や、画像などのグラフィックスについて記述します。５行目の drawString メソッドを実行し、文字列「Hello World!」を x 方向に０ドット、y 方向に１２ドットの位置に表示します。

Eclipse 等で実行すると下の図のようにアプレットビューアが起動し、アプレットが実行されます。

実際に Web ブラウザ上でアプレットを動作させるため、メモ帳などのテキストエディタを用いて次のような HTML ファイルを作成します。

HTML ファイル：hello.html

```
1  <html>
2  <head><title>アプレット</title></head>
3  <body>
4  <h1>Hello アプレット</h1>
5  <applet code="HelloApplet.class" width=100 height=100></applet>
6  </body>
7  </html>
```

５行目の applet タグでアプレットのファイル名とサイズを指定しています。ここでは縦１００ドット×横１００ドットでサイズを指定しています。

作成したHTMLファイルと同じフォルダ(ディレクトリ)に「HelloApplet.class」(「HelloApplet.Java」のワークスペース内のbinフォルダにあります)をコピーし、HTMLファイルを開くと次のように表示されます。

【課題1】今日の運勢を表示するおみくじアプレットを作成しなさい。

実行結果

※アプレットビューアによる表示

プログラム例

```
1  import java.applet.Applet;
2  import java.awt.Graphics;
3  import java.util.Random;
4  public class Omikuji extends Applet {
5          Random rnd = new Random();
6          String str[] = {"大吉","吉","中吉","小吉","末吉","凶"};
7          int i = rnd.nextInt(str.length);
8          public void paint(Graphics g){
9                  g.drawString("今日の運勢は" + str[i] + "です。",0,12);
10         }
11 }
```

乱数を生成するためにRandomクラスを使用しています。５行目でRandomクラスのインスタンスを作成し、７行目のnextIntメソッドで０から始まり「引数－１」を超えない範囲の整数を返します。ここで、引数はlengthメソッドを用いて、文字列型配列strの要素数を取得しています。

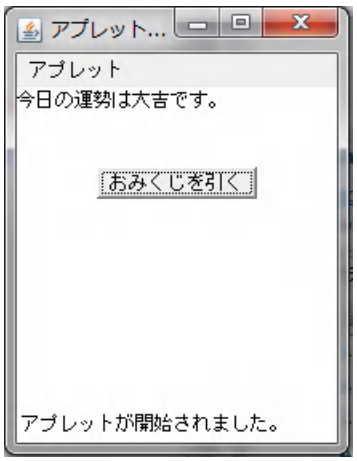
Randomクラスを使用するために、３行目でRandomクラスをインポートしています。

【課題2】今日の運勢を表示するボタンつきおみくじアプレットを作成しなさい。

実行結果

ボタンクリック前

ボタンクリック後



プログラム例

```
1  import java.applet.Applet;
2  import java.awt.Button;
3  import java.awt.Graphics;
4  import java.awt.event.ActionEvent;
5  import java.awt.event.ActionListener;
6  import java.util.Random;
7  public class Omikuji2 extends Applet implements ActionListener {
8          Random rnd = new Random();
9          String str[] = {"大吉","吉","中吉","小吉","末吉","凶"};
10         String mes = "";
11         Button b = new Button("おみくじを引く");
12         inti;
13         public void init(){
14                 setLayout(null);
15                 add(b);
16                 b.setBounds(50,50,100,20);
17                 b.addActionListener(this);
18         }
19         public void paint(Graphics g){
20                 g.drawString(mes,0,12);
21         }
22         public void actionPerformed(ActionEvent e) {
23                 i = rnd.nextInt(str.length);
24                 mes = "今日の運勢は" + str[i] + "です。";
25                 repaint();
26         }
27 }
```

ボタンの表示

ボタンを表示するにはButtonクラスを利用します。Buttonクラスを利用するために、２行目でButtonクラスをインポートし、１１行目でオブジェクトを生成しています。

initメソッドの処理

１３行目～１８行目の init メソッドにはアプレットが読み込まれたときに１度だけ行う処理を記述します。アプレットに対する初期化の処理などを行います。

１４行目の setLayout メソッドでアプレットの画面上のレイアウトを指定します。引数に null を指定することで、ボタンなどのオブジェクトをオブジェクトごとに座標や大きさを自由に設定することができます。

１５行目のadd(b)で画面上にボタンを追加し、１６行目のb.setBounds(50,50,100,20)でボタンの場所（x方向に５０、y方向に５０）と大きさ（幅１００×高さ２０）を指定しています。

イベント処理

ボタンを表示するだけではなく、ボタンがクリックされたときに特定の処理を行うようにイベントリスナの登録を行います。ボタンが押されたときのイベントの種類はActionEventというクラスで、イベントリスナを登録するためにaddActionListenerメソッドが用意されています。

１７行目のb.addActionListener(this)で表示したボタンにイベントリスナの登録を行っています。

ボタンがクリックされたときの処理を２２行目～２６行目のactionPerformedメソッドに記述します。２５行目のrepaint()メソッドで画面に表示される内容を更新します。

これらのイベント処理を行うために、４行目～５行目で ActionEvent クラスと ActionListener クラスをインポートし、７行目のクラス宣言で implements ActionListener と記述して ActionListener インターフェースを組み込む必要があります。

さまざまなイベント処理

課題２では ActionListener インターフェースを用いたイベント処理を行いましたが、マウスイベントやキーボードからの入力など、さまざまなイベント処理を行うことができます。一例を示しておきますので参考にしてください。

①ボタンがクリックされたときのイベント処理

インターフェース	ActionListener
インポートするクラス	java.awt.event.ActionEvent; java.awt.event.ActionListener;
リスナの登録	addActionListener
呼び出されるメソッド	
actionPerformed(ActionEvent e)	ボタンがクリックされたときの処理を記述します。

②キーボード操作によるイベント処理

インターフェース	KeyListener
インポートするクラス	java.awt.event.KeyEvent; java.awt.event.KeyListener;
リスナの登録	addKeyListener
呼び出されるメソッド	
keyPressed(KeyEvent e)	キーが押されたときの処理を記述します。
keyReleased(KeyEvent e)	キーが離された時の処理を記述します。
keyTyped(KeyEvent e)	キーが押されたときの処理を記述します。(主に文字キー)

③マウス操作によるイベント処理

インターフェース	MouseListener
インポートするクラス	java.awt.event.MouseEvent; java.awt.event.MouseListener;
リスナの登録	addMouseListener
呼び出されるメソッド	
mouseClicked(MouseEvent e)	マウスのボタンがクリックされたとき(マウスのボタンを押して離したとき）の処理を記述します。
mouseEntered(MouseEvent e)	マウスカーソルがアプレットの上に乗ったときの処理を記述します。
mouseExited(MouseEvent e)	マウスカーソルがアプレットから離れたときの処理を記述します。
mousePressed(MouseEvent e)	マウスのボタンが押されたときの処理を記述します。
mouseReleased(MouseEvent e)	マウスのボタンが押された状態から離されたときの処理を記述します。

④KeyListener を使ったプログラム例

```
 1 import java.applet.Applet;
 2 import java.awt.Graphics;
 3 import java.awt.event.KeyEvent;
 4 import java.awt.event.KeyListener;
 5 public class KeyTest extends Applet implements KeyListener {
 6         String str;
 7         public void init(){
 8                 str="";
 9                 addKeyListener(this);
10         }
11         public void paint(Graphics g){
12                 g.drawString(str, 0, 12);
13         }
14         public void keyPressed(KeyEvent e) {}//使用しなくても記述する必要があります
15         public void keyReleased(KeyEvent e) {
16                 str="";
17                 repaint();
18         }
19         public void keyTyped(KeyEvent e) {
20                 str=e.getKeyChar()+"キーが押されました";
21                 repaint();
22         }
23 }
```

文字キーなどが押されたときにどのキーが押されたかを表示するアプレットです。文字キー以外のキーが押された場合は何も処理が行われません。

２０行目の getKeyChar メソッドで押されたキーの文字を取得しています。

平成24年度 商業教育改善委員会

委　員　長	宿南　敏行	県立神戸商業高等学校	校　　長
副委員長	長谷川　義人	県立姫路商業高等学校	校　　長

- javaグループ

委　員	小柴　直樹	県立神戸商業高等学校	主幹教諭
〃	山本　義史	県立神戸商業高等学校	教　　諭
〃	國澤　　剛	県立神戸商業高等学校	教　　諭
〃	細川　宗一郎	県立神戸商業高等学校	教　　諭
〃	清川　光慶	市立神港高等学校	教　　諭
〃	古川　裕士	伊丹市立高等学校	教　　諭
〃	北川　欽一	県立龍野北高等学校	教　　諭
〃	鬼頭　妙子	県立姫路商業高等学校	教　　諭
〃	福谷　千佳	県立姫路商業高等学校	教　　諭

- 事　務　局

奥村　裕幸	県立神戸商業高等学校	教　　諭
貞松　敏夫	県立神戸商業高等学校	教　　諭