

# 15章解答

1. 次の表は力士、と競艇選手の身長と体重に関する条件です。身長と体重を入力してどちらの職業に就業可能か、あるいはそれ以外かを表示するクラス (プログラム) を作成しなさい。

	身長	体重
力士	173cm 以上	75kg 以上
競艇選手 (18 歳未満)	168cm 以下	52kg 以下

クラスには次の 2 つのメソッドを作成しなさい。

## A. whichJob メソッド

- ・ 仮引数に身長と体重を受け取る
- ・ 力士の条件を満たしている時は 1、競艇選手の条件を満たしている時は 2 を返す。どちらにも当てはまらない時は 0 を返す

## B. main メソッド

- ・ キーボードをタイプして身長(cm)と体重(kg)を入力し、それを whichJob メソッドに渡して適合する職業の種類を数値(0,1,2)で得る
- ・ 職業の種類が 0 なら「一般」、1 なら「力士」、2 なら「競艇選手」と表示する

【解答欄】 25 点

```
import lib.Input;
public class P7{
    public static void main(String args[]) {
        double h=Input.getDouble("身長");
        double w=Input.getDouble("体重");
        int ans=whichJob(h, w);
        if(ans==0)        System.out.println("一般");
        else if(ans==1)    System.out.println("力士");
        else               System.out.println("競艇選手");
    }
    public static int whichJob(double h,double w){
        int n=0;
        if(h >= 173 && w >= 75)        n=1;
        else if(h <= 168 && w <= 52)    n=2;
        return n;
    }
}
```

# 15章解答

《解説》

「料金を求める」処理をメソッドにして、main の処理を簡単にします。

このように、ある程度まとまった処理をメソッドにする方法は、プログラミングではしばしば使います。

特に難しくはない問題ですが、この中の whichJob メソッドでは

```
if(h >= 173 && w >= 75)      return 1;
else if(h <= 168 && w <= 52)  return 2;
else                          return 0;
```

としても構いません。

しかし、解答では if 文では n の値を変更するだけで、最後にひとつだけ return 文があり、それで n の値を返します。その理由は、メソッドの出口はひとつだけにした方がデバッグで値のチェックが簡単になるからです。単純な問題ではそこまで気を使うことはないのですが、日ごろからそのような心がけを持つことが大切です。

なお、whichJob メソッドの戻り値を if 文でなく switch 文で処理するとどうなるか考えてみてください。switch( whichJob(h, w) ){ …… } というスタイルです。

# 15章解答

2. 以下は 7 月における 2 つの航空会社の成田ーバンクーバー間の運賃と燃油サーチャージの表です。これから、入力した出発日に応じてどちらか費用の安い航空会社を選択するプログラムを作成します。以下に示す 3 つのメソッドを持つクラス(プログラム)を作成しなさい。

表の見方：

A 社では現在から 7/14 および 7/15~7/24 の期間は燃油サーチャージは 12,000 円です。しかし、7/25 以降は 16,000 円になります。これに対して B 社は現在から 7/14 までは燃油サーチャージは 12,000 円ですが、7/15~7/24 および 7/25 以降は 15,000 円です。

	運賃	燃油サーチャージの改訂日		
		現在	7/15	7/25
A 社	100000	12000	12000	16000
B 社	98000	12000	15000	15000

## A. airfareA メソッド

- ・ 出発日を `int` の仮引数に受け取る
- ・ A 社の出発日に応じた運賃と燃油サーチャージの合計金額を計算して返す

## B. airfareB メソッド

- ・ 出発日を `int` の仮引数に受け取る
- ・ B 社の出発日に応じた運賃と燃油サーチャージの合計金額を計算して返す

## C. main メソッド

- ・ キーボードをタイプして出発日を入力する
- ・ 出発日を実引数にして A 社と B 社の運賃をもとめる
- ・ どちらが易いか比較して、安い方の会社名と金額を表示する。ただし、同額ならば会社名ではなく「同額」表示する

# 15章解答

【解答欄】 25 点

```
import lib.Input;
public class P4 {
    public static void main(String[] args) {
        int date = Input.getInt("出発日");
        int a      = airfareA(date);
        int b      = airfareB(date);
        if(a<b){
            System.out.println("A 社 : " +a+"円");
        }else if(a==b){
            System.out.println("同額:" +a+"円");
        }else{
            System.out.println("B 社 : " +b+"円");
        }
    }
    public static int airfareA(int date){
        int a=100000;
        if(date<25){
            a+=12000;
        }else{
            a+=16000;
        }
        return a;
    }
    public static int airfareB(int date){
        int b=98000;
        if(date<15){
            b+=12000;
        }else{
            b+=15000;
        }
        return b;
    }
}
```

《解説》

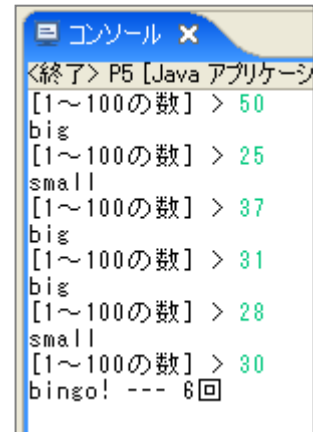
「料金を求める」処理をメソッドにして、main の処理を簡単にします。

このように、ある程度まとまった処理をメソッドにする方法は、プログラミングではしばしば使います。

# 15章解答

3. 数当てゲームを作成します。コンピュータは **1~100** の間のどれかの数をあらかじめ決めて設定します。挑戦者が予想する数を入力すると、大きすぎれば **big** と表示し、小さすぎれば **small** と表示します。挑戦者は数を当てるまで予想する数を繰り返し入力しなければなりません。入力するたびにコンピュータがその数を調べて **big** か **small** を表示します。

ただし、入力した数がコンピュータがあらかじめ設定していた数と合致すると **bingo!** と表示し、それまでの試行回数を右図のように表示してゲームは終了となります。



```
<終了> P5 [Java アプリケーション]
[1~100の数] > 50
big
[1~100の数] > 25
small
[1~100の数] > 37
big
[1~100の数] > 31
big
[1~100の数] > 28
small
[1~100の数] > 30
bingo! --- 8回
```

次の 3 つのメソッドを作成して数当てゲームプログラムを作成しなさい。

## A. number メソッド

- ・ 仮引数はなし
- ・ 乱数を使って **1** から **100** の間のどれかの数を作成し **r** に代入する
- ・ 戻り値として **r** を返す

## B. compare メソッド

- ・ 2 つの整数を仮引数 **m**, **n** に受け取る
- ・ **n>m** ならば **"big"** と表示する
- ・ **n<m** ならば **"small"** と表示する
- ・ **n==m** ならば **"bingo"** と表示する
- ・ 戻り値はない

## C. main メソッド

- ・ **number** メソッドを呼び出してコンピュータ側の数を変数 **num** に設定する
- ・ 試行回数 **n=0** としておく
- ・ 挑戦者の入力した数を代入する変数 **v** を宣言しておく
- ・ **num==v** でない間、**1), 2), 3)** を繰り返し実行する
  - 1) 数値の入力
  - 2) 比較(**compare** メソッド)
  - 3) 試行回数のカウントアップ
- ・ 合致して繰り返しが終了したら試行回数を表示する (これで終了)

<ヒント>

# 15章解答

※1～100 のどれかの値を乱数として得る方法は、練習 12-1 の問題 2 を参考にしてください

※繰り返し構文は **do-while** を使うとよいでしょう

【解答欄】 25 点

```
import lib.Input;
public class P5 {
    public static void main(String[] args){
        int num = number();
        int n =0;
        int v;
        do{
            v=Input.getInt("1～100 の数");
            compare(v, num);
            n++;
        }while(num!=v);
        System.out.println(" --- "+n+"回");
    }
    public static void compare(int a, int b){
        if(a>b){
            System.out.println("big");
        }else if(a==b){
            System.out.print("bingo!");
        }else{
            System.out.println("small");
        }
    }
    public static int number(){
        int r = (int)(Math.random*100)+1 ;
        return r;
    }
}
```

《解説》

これも3つのメソッドの組み合わせ方を見る問題です。この問題で作成するメソッドはそれぞれ機能が違います。main から number() と compare() を呼び出すことにより、main() での処理が分かりやすくなります。まとまった処理を小さな処理に分けてメソッドとし、全体の機能を組み立てる典型的な例です。

# 15章解答

4. 次のようなクラスがある。

(1) ①②に入る適語を書きなさい。

(2) ③に書いてもコンパイルエラーにならないものを選択肢から選びなさい (複数)。

```
public static void main(String[] args){
    byte a = myCalc ("Ok", 3.1 );
    System.out.println(a);
}
public static ① myCalc ( ② ){ // 仮引数名は a,b とする
    .....
    .....
    return ③;
}
```

③の選択肢

- a. -19/3
- b. 'a' + 1
- c. 512
- d. (byte)16.3
- e. 空白 (何も指定しない)

《解説》

青い部分から戻り値は byte 型であることがわかります。③は byte 型か、byte 型に自動型変換される値が入ります。a、b、が該当し、また d も byte にキャストされているので該当します。ただ、C の 512 は byte 型の値の範囲 (-128~127) を超えるので該当しません。

黄色の部分から引数の並びがわかります。(String a, double b)となっていなければなりません。

【解答欄】 @5×3=15 点

① byte	② String a, double b	③ a, b, d
--------	----------------------	-----------

5. 次のようなメソッド **foo** があるとき、これを呼び出す書き方として誤っているものはどれか番号で答えなさい (複数ある)。

```
public static double foo(double x, int n){
    .....
    .....
    return x;
}
```

10 点

- A. int n = foo(5.5, 10);
- B. double x = foo(10.5);
- C. double x = foo(152, 10);
- D. double x = foo(1.5, 10, 0);
- E. int n = (int)foo(12/3.0, 5/4);
- F. char c = (char)foo(5.5F, 10);

【解答欄】

A, B, D

# 15章解答

## 《解説》

アンダーラインを引いたところが、エラーになるところです。A は double の戻り値を int で受けています。B と D は引数の数が 2 個ではありません。

なお、 $5/4$  は一見 double になりそうに見えますが、5 も 4 も整数なので計算結果は整数になります。つまり 1 になることに注意してください。