

第 52 回検定 (平成 27 年 1 月 18 日) 全商情処プロ部門 Java 1 級解説

【 7 】

概要

ちょっと変わったオブジェクト指向で設計されたクラスやメソッドを用いて順位付けをするプログラム。

詳細

1 級の問題は、まずはクラスやメソッドを全てチェックするほうが良い。具体的には、クラスやメソッドの先頭行をマーカーでチェックしよう。たとえばこんな感じ。

```
public class ShoInfo{
    private ...
    private ...
    public ShoInfo(String name){
        this.name = ...
        for(int j = ...
```

こうすれば、パッと見て「クラス!」とか「コンストラクタ!」とか「メソッド!」とか把握できます。「木を見て森を見ず」ではいけません。まずは、プログラムの概略を把握すること。

この問題は `ShopInfo` クラスをインスタンス化して、`ShopInfo` クラスのメソッドを利用して売上データを集計します。そして、`ShopInfo` クラスのメソッドを利用して順位をつけ、表示をします。また、`ShopInfo` クラスには `static` なフィールドが複数あり、これらはみんなが仲良く共通で使えます。インスタンス化する必要はありません。難しく言うと、`static` なものはインスタンスの内部にあるのではなく、クラスそのものに実体があるのだ!ということなのだ。

(1) はコンストラクタの内部にあります。つまり初期化の処理です。1つ上の `total[j] = 0` を見ると、視線が処理条件の 3 に移るので、その流れで何を初期化するのかなんとなくわかるでしょう。

(2) は「`sales =`」となっているので売上高を求めていることはわかるでしょう。そうなると、売上高は数量×単価で求める。ちなみに、`salesTotal` メソッドの引数の 1 つでもある `quantity` とは日本語で言うところの「数量」。つまり売上数を表しています。これからのプログラマーには英語も必要なのです。プログラミングと英語が両方学べる `Java` って素敵ですね。

(3) はメソッド名からも、配列の名前からも順位付けであることは明白です。順位付けの問題は、まず最初に「何順に順位を付けるのか」を把握すること。昇順と降順では、答えるべきことが真反対になりますよ。処理条件を見たら「○順に順位をつける」と必ず書いていますからね。それを見つけたら、あとは (3) の 1 つ下を見ます。m ではなく n の方に 1 プラスしています。つまり、順位を 1 プラスする、ということは n の方が順位が劣っているということ。あとは自分で解いて下さい。

(4) はコンストラクタの呼び出しです。これは、`ShopInfo` クラスのコンストラクタ部分を見ると、答えが一瞬でわかってしまいます。「コンストラクタって何?おいしいの?」と言う人は、2 級からやり直しましょう。

(5) が含まれている `while` は出力のためのものです。(5) の前後で `outName` メソッドや `outTotal` メソッドが呼び出されているので、なんとなく出力、ということは理解できるでしょう。ちなみに、`outName` メソッドは COBOL という見出行、`outTotal` メソッドは合計行といったところでしょうか。そう考えても「肝心の明細行がないじゃないか！」ということに気づきます。ええ、気づくとも。そこで、`outName` メソッドや `outTotal` メソッドを呼び出している命令を参考に、`outDetails` メソッドを呼び出してあげましょう。ちなみに `Detail` とは「詳細」や「細部」という意味です。有名な言葉に「神は細部に宿る」というものがあります。God is in the details です。Java って英語の勉強も出来てお得な言語ですね。

ただ、(5) を解く上で注意しなければならないのは `outDetails` メソッドは引数を取っている、ということ。ShopInfo クラスの `outDetails` メソッドを見てみると、`outDetails(int j)` となっています。つまり、何かしら `int` な数値を渡してあげないとはいけません。大切なので何度も言いますが、こうやってメソッドに引数を「渡す」という考え方はよく理解して下さいね。

そして、配列のカッコは `[]`、引数を指定するカッコは `()` です。この 2 つのカッコを混同して間違えないようにしましょう。ちなみに私は間違えました。

解答

(1)	<code>rank[j] = 1</code>
(2)	<code>quantity * sweetsPrice[sweetsCode]</code>
(3)	<code>total[m] > total[n]</code>
(4)	<code>name[i]</code>
(5)	<code>shop[code].outDetails(j)</code>

この検定解説プリントを使用している方がおられましたらご感想を下さい。というのも作成してから一度も、誰からも、「使ったよ」という連絡がありません。必要とされていないなら作成を止めようと考えています。ご感想お願いします。