

第 56 回検定(H29.01.29)全商情処プロ部門 Java 2 級 解説

【 7 】

特徴

データの集計・平均	一次元配列の利用	メソッドの呼び出し
-----------	----------	-----------

概要

店舗ごとに Tenpo クラスのインスタンスを生成する。その後、データを読み込みながらインスタンスの keisan メソッドを使い集計する。ユーザーが入力した番号に応じてインスタンスを探し、インスタンスの hyoji メソッドで出力する。

集計と表示の機能をメソッドとしてクラスに持たせる、という第 53 回検定以降続く 2 級の典型的な問題である。また、各々の問いの難易度も低く非常に簡単な問題である。

詳細 (プログラムの流れを追って説明)

- 1 main メソッドからスタートする。スタートすると、Tenpo クラスのインスタンスを 3 つ生成している。その時に "オンラインショップ"、"東京店"、"大阪店" という文字列を渡してインスタンス化している。インスタンス化するとコンストラクタが実行される。

↓ Tenpo クラスのコンストラクタへ処理が移動

- 2 クラス名と同名の Tenpo メソッドがコンストラクタである。上記 1 で受け取った文字列をインスタンス変数 tenMei に代入している。受け取った変数名も tenMei なのでインスタンス変数は this.tenMei と区別している。ちなみに tenMei のように単語の要素間を大文字にして区切る変数名のつけ方をキャメルケースと言います。

↓ main メソッドの続きに戻る

- 3 While 部分でファイルを読み込んでカンマで分割し変数に代入する。そして、その読み込んだ変数を使ってインスタンスの keisan メソッドを呼び出ししている。

↓ Tenpo クラスの keisan メソッドへ処理が移動

- 4 keisan メソッドでは受け取った値を使って計算して配列に集計している。(1) は数量の集計なので単純に答えがわかるでしょう。処理条件の 5 番でも、ご丁寧に「配列 kin に集計する」と同じように「配列 su に数量を集計する」と書いているので、処理条件とソースコードを対応させても (1) は容易に解けます。

- 5 このようにデータがなくなるまで While 文で上記の 3 と 4 を繰り返す。

- 6 データを読み終えると、次は店舗コードを入力して該当するデータを出力する処理へと移る。(2つ目の while 文のこと)
- 7 ここで第 56 回検定 2 級が一番難しい問題(5)の登場です。毎回 5 問中 1 問出題されるオブジェクト指向の理解を問う問題です。hyoji メソッドを呼び出しているのが、hyoji メソッドを持っているのは何だ?という事です。hyoji メソッドは Tenpo クラスが持っているけど、Tenpo クラスはインスタンス化しないと使えない。じゃあ、Tenpo クラスをインスタンス化したものは何か?という main メソッドすぐ下の new 部分です。この new 部分でインスタンス化している。これがポイントです。インスタンス化して、その参照を配列 tp に代入しています。つまり、配列 tp 経由でインスタンスのメソッドを呼び出す、ということです。これで(5)がわかったかな。

↓ Tenpo クラスの hyoji メソッドへ処理が移動

- 8 いきなり for 文で(2)が出てくるけど、これは第 54 回検定で問われた個所と同じ。文法を知っていれば解ける簡単な問題。なぜ、「i = 0」ではなくて「i = 1」なのかというと、配列の 0 番目を使わずに 1 番目から使う、という問題だからです。本来であれば 0 番目から使うのですが、1 番目の方が店舗コードと上手に対応するしわかりやすい、という配慮でしょう。そして(3)と(4)があります。しかし、これは説明の必要がないくらい簡単なので省略です。

余談 1

Tenpo クラスの中にある配列 bunMei は"アレンジ"とかの値ではなく、"アレンジ~~~~"と後に空白がくっついています。これは出力で縦にキレイに揃えるためです。本来は値そのものと出力は切り離れた処理の方が良いです。ただ printf で上手く縦に揃わない苦肉の策でしょう。そもそもコンソールに出力すること自体に無理があるのかもしれない。

余談 2

hyoji メソッドに「String hantei;」があります。その下の処理で、hantei は「◎」「○」「(空白)」のいずれかの値を必ず取るので、ここでは「String hantei = "" ;」と明示的に初期化する方法もあります。この辺りはプログラムを作る人のセンス、というか趣味の問題ですね。色々な方法があるからプログラミングは面白いのです。

解答

(1)	su[bunCode] + suryo	易
(2)	i = 1	易
(3)	heikin >= 6000	易
(4)	hantei	易
(5)	tp[nyuCode]	難