

第 53 回検定 (H27.09.27) 全商情処プロ部門 Java 1 級 解説

【 7 】

概要

全体的にバランスの取れた良質なオブジェクト指向の問題。1 級過去問題の中では非常に簡単な部類。オブジェクト指向に関する問題は 5 問中 2 問。良くも悪くも「オブジェクト指向 2 ~ 3 問、他に簡単な問題 2 ~ 3 問」という傾向も固定化されてきている。巨大な 1 つのクラス EventInfo に様々なメソッドを持たせて、集計しつつインスタンスメソッドに作業させる、というパターン。このパターンも、良くも悪くも第 51 回検定以降固定化されてきている、と考えてよいかもしれない。

オーソドックスなオブジェクト指向の問題ながら、順位付けやソートしているかどうかフラグを立てる等、ちょっとずつ独自性を打ち出した問題となっている。

詳細

まずはクラスとメソッドを見る。クラスが 2 つしかなく、EventInfo クラスの中には、コンストラクタと addPoint メソッド、getFlag メソッド、sort メソッド、printPoint メソッドがある。メソッド名からもメソッドの用途がわかるので理解しやすい。

次にインスタンス化されている場所を見る。main メソッドの直後の For でインスタンス化されている。(4) の箇所である。これもわかりやすい。

処理内容を見てみると、処理条件 5 より、並べ替えと順位付けということがわかる。しかし、配列 ranking は 3 つの要素しかなく「あれ、順位付けがなんか変だな」と思いながら詳しくプログラムを見ていきましょう。

(1) は簡単。ポイントを加えるので、addPoint メソッドなのです。ただし、ここは単に「point」では誤答です。わかりにくいですが処理条件 4 の「なお、」でデータが複数ある場合もあると書いています。ポイントを代入するのではなく、「+ point」で追加していきましょう。

(2) はバブルソートです。基本的な問題です。(2) の下の交換処理で、隣りと入れ替えていることを考えるとバブルソートだな^^とわかります。降順に並べ替えるので、左が大きくないといけません。よって、左が小さかったら交換です。

(3) は難しいです。しかし、オブジェクト指向やとは本質的には関係のない所での難しさです。これは同順位を処理しています。例えば、実行結果の 2 年です。1 位が 2 つある、とか。(3) の上の if は、「ポイントが違うか、もしくは 1 番目か」という条件です。ポイントが違う、がわかりやすいです。ポイントが違うなら順位は変わる。順位は増えていきます。最大で 6 位までです。こう考えると、変数 i が順位に使えることが分かります。変数 j は何かというと、3 つあるので 1 年 ~ 3 年の学年を表しています。学年ごとに同順位処理を行っているということ。6 回ループする中に 3 回のループが入っている、という大局を見てから考えるとよいかな。(3) は難しいですが、特殊な問題なのでできなくても気にしなくていいですよ。

(4) は普通のオブジェクト指向の問題です。ただ、(4) の 4 行上の所と (4) の違いの理解は重要です。(4) の 4 行上では、`private static EventInfo[] event = new EventInfo[9];` となっています。private static は本質的にどうでもよいので無視します。

`EventInfo[] event = new EventInfo[9];` というのは、一次元配列の event を要素 9 つで生成した、ということです。つまり、単なる要素 9 つの配列を作った、ということ。じゃあ `EventInfo[]` とか、`EventInfo[9]` とは何か、というと・・・`Int[]` とか `Int[9]` と同じ意味です。Int 型ではなくて EventInfo 型を代入する配列だよ～、ということを宣言しています。こっちの new はインスタンス化の new ではなくて、単なる配列生成の new です。

(4) ではその 9 つの要素がある配列 event に、EventInfo のインスタンスの参照を代入しています。こっちの new はインスタンス化の new で、この new でコンストラクタが呼ばれます。コンストラクタを呼ぶときにイベント名 (name[i]) を渡しながらか呼び、そしてコンストラクタを実行している、ということです。

(5) はオブジェクト指向の内容ですが簡単ですね。こういう変数の後にドットでつながれた記述だとインスタンスのメソッドしかありえないので、自然と答えはメソッドになります。

なお、クラス EventInfo の上から 3 つ目のメソッド getFlag はアクセッサと呼ばれるものです。sortFlag というフィールドの値を GET するためだけのメソッドで Getter と呼ばれます。一般的にはアクセッサの命名規則は、フィールド名の前に Set や Get をつけますので、厳密には getFlag ではなくて getSortFlag がメソッド名としては望ましいでしょう。また、フラグは通常 True か False の値を保持します。そう考えると sortFlag は int 型ではなく、boolean 型 (2 級範囲) が望ましいです、というかそのほうが可読性が高まりますし、True か False 以外を代入できませんのでミスも減ります。

あと、Java は大文字小文字は区別します。今回だと p が小文字と大文字の区別がつきにくいので、採点する人のことを考えて、小文字なら p、大文字なら **P** と明確に区別して書きましょうね。

解答

(1)	<code>classPoint[i][j] + point</code>
(2)	<code>classPoint[i][k] < classPoint[i][k + 1]</code>
(3)	<code>ranking[j] = i</code>
(4)	<code>event[i]</code>
(5)	<code>printPoint()</code>