

SA31 課題 4

課題名称

抽象クラスとインターフェースの確認

課題区分

訓練

納期

2025 年 12 月 17 日(水)

12/17 は課題確認の時間とする。

内容

あらかじめ用意された抽象クラスとインターフェースを利用した以下のプログラムを作成する。

パッケージと作成クラス

作成するクラスは以下の 5 クラス。パッケージは
local.hal.sa31.java.lesson3.asgmt
とし、JavaLesson3 プロジェクト内に作成する。

- JudgementRule60
- JudgementRule70
- MikeJudgement
- CindyJudgement
- DoJudgement

上記 5 クラスとは別に、以下の 3 クラスがある。これらは、別紙を参照し、写経する。

- ScoreHolder.java
- JudgementRule.java
- AbstractJudgement.java

作成するクラスの仕様

作成する 5 クラスは以下の仕様とする。

JudgementRule60

JudgementRule を実装(implements)したクラスとし、合計点の 6 割で合格とするコードが記述されている。

JudgementRule70

JudgementRule を実装(implements)したクラスとし、合計点の 7 割で合格とするコードが記述されている。

MikeJudgement

AbstractJudgement を継承(extends)したクラスとし、名前は「マイク」、SA が 70 点、PH が 56 点、JS が 53 点とする。

CindyJudgement

AbstractJudgement を継承(extends)したクラスとし、名前は「シンディー」、SA が 60 点、PH が 68 点、JS が 82 点とする。

DoJudgement

実行クラス。MikeJudgement に JudgementRule60 を適用して、judge()を実行する。続けて、CindyJudgement に JudgementRule70 を適用して、judge()を実行する。

実行結果

実行結果は、以下を参照。

マイクさんの総合計は 179 点で、結果は不合格です。

シンディーさんの総合計は 210 点で、結果は合格です。

提出物と提出先

1. アプリのデモ → 講師まで直接見せにくる
デモ時にソースコードも確認する。
2. 以下の 5 ファイル → Teams の課題
zip 化する必要はない。
 - JudgementRule60
 - JudgementRule70
 - MikeJudgement
 - CindyJudgement
 - DoJudgement

```
1 package local.hal.sa31.java.lesson3.asgmt;
2
3 /**
4 * SA31 Java文法 Lesson3 課題4
5 *
6 * 学生の点数を保持するクラス。
7 *
8 * @author Shinzo SAITO
9 */
10 public class ScoreHolder {
11     /**
12      * 学生名を表すフィールド。
13      */
14     private String _name;
15     /**
16      * SA科目の点数を表すフィールド。
17      */
18     private int _sa;
19     /**
20      * PH科目の点数を表すフィールド。
21      */
22     private int _ph;
23     /**
24      * JS科目の点数を表すフィールド。
25      */
26     private int _js;
27
28     /**
29      * コンストラクタ。
30      *
31      * @param name 学生名。
32      * @param sa SA科目の点数。
33      * @param ph PH科目の点数。
34      * @param js JS科目の点数。
35      */
36     public ScoreHolder(String name, int sa, int ph, int js) {
37         _name = name;
38         _sa = sa;
39         _ph = ph;
40         _js = js;
41     }
42
43     /**
44      * 科目の合計点を取得するメソッド。
45      *
46      * @return 合計点。
47      */
48     public int getTotalScore() {
49         return _sa + _ph + _js;
50     }
51
52     /**
53      * 学生名を取得するメソッド。
54      *
55      * @return 学生名。
56      */
57     public String getName() {
58         return _name;
59     }
60 }
61
```

```
1 package local.hal.sa31.java.lesson3.asgmt;
2
3 /**
4 * SA31 Java文法 Lesson3 課題4
5 *
6 * 合否判定ロジックインターフェース。
7 *
8 * @author Shinzo SAITO
9 */
10 public interface JudgementRule {
11     /**
12      * 合否を判定するメソッド。
13      *
14      * @param totalScore 合計点。
15      * @return 合格ならtrue、不合格ならfalse。
16      */
17     boolean judge(int totalScore);
18 }
19
```

```
1 package local.hal.sa31.java.lesson3.asgmt;
2
3 /**
4  * SA31 Java文法 Lesson3 課題4
5  *
6  * 学生の点数と合否判定ロジックを利用して合否判定を行うクラス。
7  *
8  * @author Shinzo SAITO
9  */
10 public abstract class AbstractJudgement {
11     /**
12      * 合否判定ロジックが記述されたオブジェクトフィールド。
13      */
14     private JudgementRule _rule;
15
16     /**
17      * コンストラクタ。
18      *
19      * @param rule 合否判定ロジックが記述されたオブジェクト。
20      */
21     public AbstractJudgement(JudgementRule rule) {
22         _rule = rule;
23     }
24
25     /**
26      * 学生の点数を表すオブジェクトを生成するメソッド。
27      *
28      * @return 学生の点数を表すオブジェクト。
29      */
30     protected abstract ScoreHolder createScoreHolder();
31
32     /**
33      * 合否結果を表示するメソッド。
34      */
35     public void judge() {
36         ScoreHolder score = createScoreHolder();
37         int totalScore = score.getTotalScore();
38         boolean result = _rule.judge(totalScore);
39         String resultStr = "不合格";
40         if(result) {
41             resultStr = "合格";
42         }
43         String resultMsg = score.getName() + "さんの総合計は" + totalScore + "点で、結果は"
44         + resultStr + "です。";
45         System.out.println(resultMsg);
46     }
47 }
```