

課題3_1班_杉谷一祝

```

1 package jp.project1.testsumlist;
2
3 /*****
4  *   試験結果集計プログラム : Studentクラス
5  *
6  *   プログラム名 : Student
7  *   概要 :       試験結果データ用
8  *   作成日付 :    2020/09/25
9  *   版数 :       1.1版
10  *   作成者(班:PL) : 杉谷一祝(1:京岡大純)
11  *   修正履歴 :    1.1版   フィールドにint型変数rankを追加
12  *   備考 :       なし
13  *   課題No :      3
14  *****/
15
16 import java.util.Arrays;
17
18 /**
19  * Studentクラス
20  */
21 public class Student {
22     /** for文カウンタ変数の初期値 */
23     public static final int ZERO          = 0;
24     /** 欠席している場合 */
25     public static final int NON_TESTER    = -1;
26     /** 欠席者の点数 */
27     public static final int NON_TESTER_SCORE = 0;
28     /** 赤点 */
29     public static final int RED_SCORE     = 25;
30
31     /** 順位 1.1版 */
32     protected int      rank;
33     /** 各科目の得点 */
34     private int []      scores;
35     /** 氏名 */
36     private String      name;
37     /** 再試験に該当するかどうか */
38     private boolean     isRetester;
39
40     /**
41      * 初期化用コンストラクタ
42      * @param name 氏名
43      * @param scores 各科目の得点
44      */
45     public Student(String name, int[] scores) {
46         /* 氏名を格納 */
47         this.name = name;
48
49         /* 得点から再試験者を判定 */
50         for (int i = ZERO; i < scores.length; i++) {
51             /* 欠席の場合 */
52             if (scores[i] == NON_TESTER) {
53                 scores[i] = NON_TESTER_SCORE;
54                 isRetester = true;
55                 continue;
56             }
57             /* 赤点の場合 */
58             if (scores[i] <= RED_SCORE) {
59                 isRetester = true;
60             }
61         }
62
63         /* 得点を格納 */
64         this.scores = scores;
65     }
66
67     /*****
68     *   メソッド(クラス)名 : getScores(Student)
69     *
70     *   メソッド名 :      getScores
71     *   概要 :          各科目の得点を取得
72     *   引数 :          なし
73     *   返却値 :        int[]型scores
74     *   備考 :          なし
75     *****/
76     /**
77      * 各科目の得点を取得
78      * @return int[]型scores
79      */
80     public int[] getScores() {
81         return scores;
82     }
83
84     /*****
85     *   メソッド(クラス)名 : getName(Student)
86     *
87     *   メソッド名 :      getName
88     *   概要 :          氏名を取得
89     *   引数 :          なし
90     *   返却値 :        String型name
91     *   備考 :          なし
92     *****/
93     /**
94      * 氏名を取得
95      * @return String型name
96      */
97     public String getName() {
98         return name;

```

```

99     }
100
101 /*****
102 *   メソッド(クラス)名 : getIsRetester(Student)
103 *
104 *   メソッド名 :      getIsRetester
105 *   概要 :          再試験に該当するかどうか
106 *   引数 :          なし
107 *   返却値 :        boolean型isRetester
108 *   備考 :          なし
109 *****/
110 /**
111  * 再試験に該当するかどうか
112  * @return boolean型isRetester
113  */
114 public boolean getIsRetester() {
115     return isRetester;
116 }
117
118 /*****
119 *   メソッド(クラス)名 : getSum(Student)
120 *
121 *   メソッド名 :      getSum
122 *   概要 :          得点の合計を算出
123 *   引数 :          なし
124 *   返却値 :        int型の得点の合計値
125 *   備考 :          なし
126 *****/
127 /**
128  * 得点合計値
129  */
130 public int getSum() {
131     return Arrays.stream(scores).sum();
132 }
133
134 /*****
135 *   メソッド(クラス)名 : toString(Student)
136 *
137 *   メソッド名 :      toString
138 *   概要 :          必要なフィールドの返却
139 *   引数 :          なし
140 *   返却値 :        String型のフィールド値一覧
141 *   備考 :          デバッグ用
142 *****/
143 /**
144  * デバッグ用
145  */
146 @Override
147 public String toString() {
148     return "Student [isRetester=" + isRetester + ", name=" + name + ", scores=" + Arrays.toString(scores) + "]";
149 }
150
151 }

```