

```

1 package tpe.collections.minions;
2
3 import tpe.collections.minions.domain.Minion;
4 import tpe.collections.minions.domain.MinionFarbe;
5 import tpe.collections.minions.domain.MinionIterable;
6
7 import java.util.HashSet;
8 import java.util.Set;
9
10 public class Main {
11     /**
12      * Gibt einen String zur Ausgabe aller errechneten Werte
13      * zurück.
14      * @param gesamtOD      Minions ohne DubLetten
15      * @param gelbOD        Gelbe Minions ohne DubLetten
16      * @param gelbeODP      Gelbe Minions ohne DubLetten in
17      *                      Prozent
18      * @param violettOD    Violette Minions ohne DubLetten
19      * @param violettODP   Violette Minions ohne DubLetten in
13      *                      Prozent
19      * @param gelbviolettODV Verhältnis Gelbe und Violette
20      * Minions
21      * @param gesamtMD    Minions mit DubLetten
22      * @param gelbMD       Gelbe Minions mit DubLetten
23      * @param gelbeMDP    Gelbe Minions mit DubLetten in
24      *                      Prozent
23      * @param violettMD   Violette Minions mit DubLetten
24      * @param violettMDP  Violette Minions mit DubLetten in
24      *                      Prozent
25      * @param gelbviolettMDV Verhältnis Gelbe und Violette
26      * Minions
26      * @param fehler       Fehler bei der Zählung
27      */
28      * @return Formatierter String mit allen wichtigen Infos
29      */
30      public static String outputString(int gesamtOD, int gelbOD,
31                                         double gelbeODP,
32                                         int violettOD, double
33                                         violettODP, double gelbviolettODV, int gesamtMD,
34                                         int gelbMD, double gelbeMDP,
35                                         int violettMD, double violettMDP,
36                                         double gelbviolettMDV,
37                                         double fehler) {
38      /*
39      * Ohne DubLetten: 1128
40      * Minions 717 gelbe (64%) und 411 violett (36%),
41      * gelb/violett 1,74
42      * -----
43      * Mit DubLetten: 1184 Minions 758 gelbe (64%) und 426

```

```

39     violett (36%),
40             * gelb/violett 1,78
41             *
42             * -----
43             * Fehler bei der Zählung: 4,7%
44             */
45         return ("Ohne Dubletten: " + gesamtOD + " Minions\n" +
46                 gelbOD
47                         + " gelbe (" + gelbeODP + "%) und " + violettOD +
48                         " violett (" + violettODP + "%), gelb/violett " +
49                 gelbviolettODV + "\n"
50                         +
51                         "-----\n" +
52                         + "Mit Dubletten: " + gesamtMD + " Minions\n" +
53                 gelbMD
54                         + " gelbe (" + gelbeMDP + "%) und " + violettMD +
55                         " violett (" + violettMDP + "%), gelb/violett " +
56                 gelbviolettMDV + "\n"
57                         +
58                         "-----\n" +
59                         + "Fehler bei der Zählung: " + fehler + "\n");
60     }
61
62     /**
63      * ermittelt Gesamtanzahl Minions
64      *
65      * @param factory
66      *
67      * @return integer die Anzahl
68      */
69     public static int getAnzahlGesamt(MinionIterable factory) {
70         int anzahl = 0;
71         for (Minion m : factory) {
72             anzahl++;
73         }
74         return anzahl;
75     }
76
77     /**
78      * Zähle gelbe Minions mit Dupletten
79      *
80      * @param factory
81      *
82      * @return integer die Anzahl
83      */
84     public static int getGelbeMinionsMD(MinionIterable factory) {
85         int anzahl = 0;
86         for (Minion n : factory) {
87             if (n.getFarbe() == (MinionFarbe.GELB)) {

```

```

81             anzahl++;
82         }
83     }
84     return anzahl;
85 }
86
87 /**
88 * Zähle violette Minions mit Dupletten
89 *
90 * @param factory
91 *
92 * @return integer die Anzahl
93 */
94 public static int getVioletteMinionsMD(MinionIterable
factory) {
95     int anzahl = 0;
96     for (Minion n : factory) {
97         if (n.getFarbe() == (MinionFarbe.VIOLETT)) {
98             anzahl++;
99         }
100    }
101    return anzahl;
102 }
103
104 /**
105 * Zähle alle Minions ohne Dupletten
106 *
107 * @param factory
108 *
109 * @return integer die Anzahl
110 */
111 public static int getMinionsOD(MinionIterable factory) {
112
113     Set<Minion> set = new HashSet<Minion>();
114
115     for (Minion n : factory) {
116         set.add(n);
117     }
118
119     return set.size();
120 }
121
122 /**
123 * Zähle alle gelbe Minions ohne Dupletten
124 *
125 * @param factory
126 *
127 * @return
128 */
129 public static int getMinionsODGelb(MinionIterable factory) {

```

```

130         Set<Minion> set = new HashSet<Minion>();
131
132         for (Minion n : factory) {
133             if (n.getFarbe() == MinionFarbe.GELB)
134                 set.add(n);
135         }
136
137         return set.size();
138     }
139
140     /**
141      * Zähle alle violetten Minions ohne DupLetten
142      *
143      * @param factory
144      *
145      * @return
146      */
147     public static int getMinionsODViolett(MinionIterable factory
148 ) {
149         Set<Minion> set = new HashSet<Minion>();
150
151         for (Minion n : factory) {
152             if (n.getFarbe() == MinionFarbe.VIOLETT)
153                 set.add(n);
154         }
155
156         return set.size();
157     }
158
159     /**
160      * Berechnet Prozentwert der Eingabe *
161      *
162      * @param gesamt 100 %
163      * @param wert wieviel prozent sind das?
164      *
165      * @return integer den Prozentsatz
166      */
167     public static double berechneProzent(double gesamt, double
168 wert) {
169         return Math.round((100.0 * wert) / gesamt);
170     }
171
172     /**
173      * Berechnet Differenz zweier Werte in Prozent
174      *
175      * @param gesamt alle zusammen
176      * @param ohneDupLetten keine Dupletten enthalten
177      *
178      * @return integer die Anzahl ohne DupLetten
179      */

```

```

178     public static double berechneFehlerQuote(double gesamt,
179         double ohneDupletten) {
180
181         double tmp = gesamt - ohneDupletten;
182
183         return Math.round((100.0 * tmp) / gesamt);
184     }
185
186     /**
187      * Ermittelt das Verhältnis von a zu b
188      *
189      * @param a erster Parameter
190      * @param b zweiter Parameter
191      *
192      * @return double das Verhältnis von a zu b
193     */
194     public static double verhaeltniss(double a, double b) {
195
196         return (double) Math.round(a / b * 100) / 100;
197     }
198
199     public static void main(String[] args) {
200
201         MinionIterable factory = new MinionIterable();
202
203         // OHNE DUBLETEN
204         int gesamtOD = getMinionsOD(factory);
205         int gelbOD = getMinionsODGelb(factory);
206         double gelbODP = berechneProzent(gesamtOD, gelbOD);
207         int violettOD = getMinionsODViolett(factory);
208         double violettODP = berechneProzent(gesamtOD, violettOD)
209         ;
210         double gelbviolettODV = verhaeltniss(gelbOD, violettOD);
211
212         // MIT DUBLETEN
213         int gesamtMD = getAnzahlGesamt(factory);
214         int gelbMD = getGelbeMinionsMD(factory);
215         double gelbMDP = berechneProzent(gesamtMD, gelbMD);
216         int violettMD = getVioletteMinionsMD(factory);
217         double violettMDP = berechneProzent(gesamtMD, violettMD)
218         ;
219         double gelbviolettMDV = verhaeltniss(gelbMD, violettMD);
220
221         //FEHLER
222         double fehler = berechneFehlerQuote(gesamtMD, gesamtOD);
223
224         System.out.println(outputString(
225             gesamtOD, gelbOD, gelbODP, violettOD, violettODP
226             , gelbviolettODV,
227             gesamtMD, gelbMD, gelbMDP, violettMD, violettMDP

```

```
223 , gelbviolettMDV,  
224             fehler  
225         ));  
226     }  
228 }  
229
```