

Aufgabe 5: Marktwage

Team-ID: 00773

Team-Name: RBBK21ItaO_Kp

Bearbeiter dieser Aufgabe:
Diyar Hasan

12. November 2021

Inhaltsverzeichnis

Lösungsidee	2
Szenarien-Beispiele	3
Umsetzung	6
Quellcode.....	8

Lösungsidee

Es gibt drei mögliche Szenarien, um die Aufgabe zu lösen,

1. die erste besteht darin, die Summe der leichteren Gewichte auf die linke Seite zu legen, um ein Gleichgewicht zu schaffen



2. Bei dem zweiten Szenario wenn nur durch die Summe leichteren Gewichte kein Gleichgewicht möglich ist, wird versucht, schwerere Gewichte auf linke Seite und leichtere Gewichte auf rechte Seite zu platzieren, um ein Gleichgewicht zu erreichen.



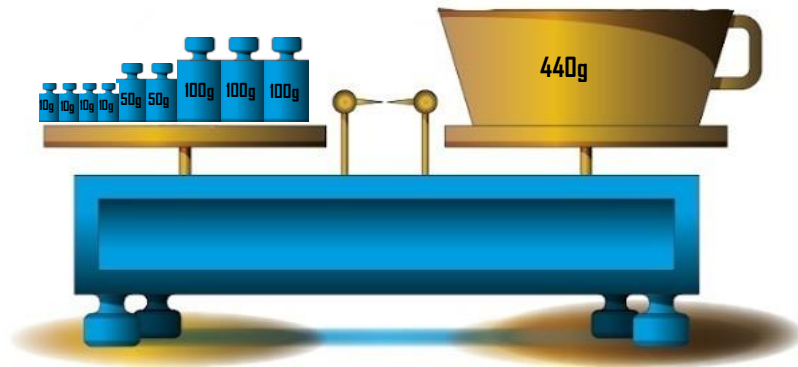
3. Die dritte basiert auf diesen beiden vorherigen möglichen Szenarien. Wenn bei den beiden Fällen kein Gleichgewicht vorhanden ist, dann sollte das schwerere oder leichtere Gewicht, das rechts am nächsten liegt, auf der linken Seite platziert werden, je nachdem welche am nächsten liegt.



Szenarien-Beispiele

Ein Beispiel für das erste Szenario:

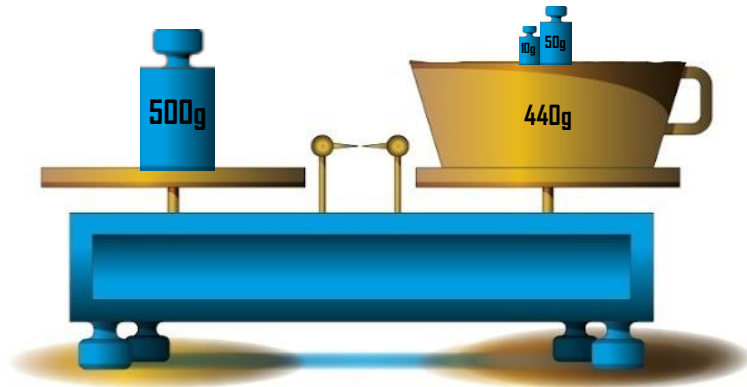
5000g, 1000g, 1000g, 1000g, 500g, 500g, 500g, 100g, 100g, 100g, 50g, 50g, 10g, 10g, 10g, 10g



Kommentar! Balance ist hier möglich, indem man leichte Gewichte auf der linken Seite packt
also $440g = 100g + 100g + 100g + 50g + 50g + 10g + 10g + 10g + 10g$

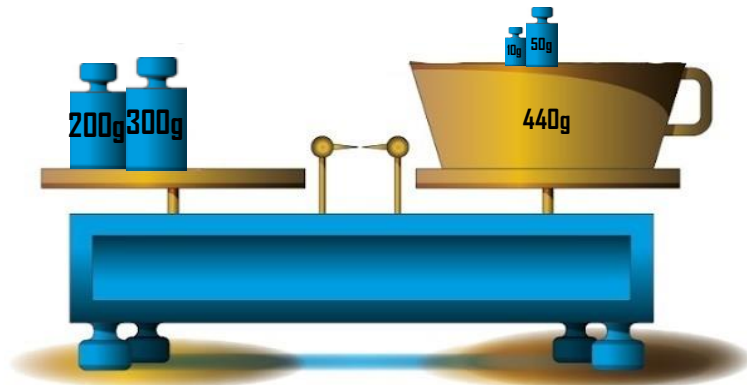
Ein Beispiel für das zweite Szenario:

5000g, 1000g, 1000g, 1000g, 500g, 500g, 500g, 100g, 100g, 100g, 50g, 50g, 10g, 10g, 10g



Kommentar! Ausbalancieren ist hier möglich, indem ein Gewicht auf linke Seite platziert wird, das schwerer ist als die rechte Seite, d. h. **500g**. Dann die Leichtgewichte **60g=50g+10g** auf die rechte Seite legen, damit in diesem Fall ein Gleichgewicht besteht.

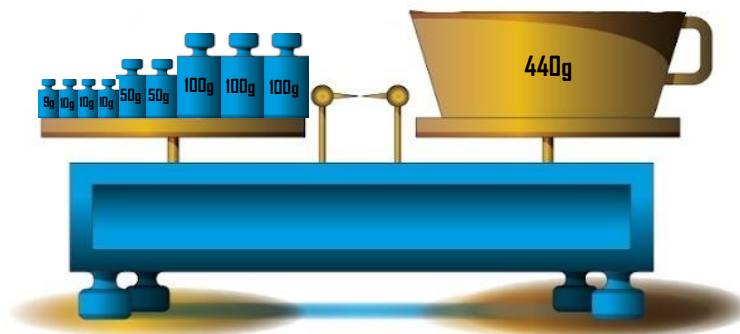
- 2.1 Aber was wenn man kein schweres Gewicht vorhanden hat, das auf linke Seite gesetzt wird wie in diesem Beispiel: 300g, 200g, 100g, 100g, 100g, 50g, 50g, 10g, 10g, 10g



Kommentar! Auch hier ist ein Gleichgewicht mit leichten Gewichten möglich, die durch eine Summe schwerer als rechte Seite sind, z.B. **300g+200g**. Dann die Leichtgewichte **60g=50g+10g** auf die rechte Seite legen, damit in diesem Fall ein Gleichgewicht besteht.

Ein Beispiel für das dritte Szenario:

5000g, 1000g, 1000g, 1000g, 500g, 500g, 500g, 100g, 100g, 100g, 50g, 50g, 10g, 10g, 10g, 13g, 9g, 6g, 6g



Kommentar! Es werden immer alle möglichen Kombinationen berechnet und die nächstgelegene als Vergleich herangezogen!!

Ein Gewicht schwerer!

500	0	0	0	0	0
445	$100g \cdot 3$	$50 \cdot 2$	$10 \cdot 3$	9	6
443	$100g \cdot 3$	$50 \cdot 2$	$10 \cdot 3$	13	0
442	$100g \cdot 3$	$50 \cdot 2$	$10 \cdot 3$	$6 \cdot 2$	0

Ein Gewicht leichter!

439	$100g \cdot 3$	$50 \cdot 2$	$10 \cdot 3$	9	0
439	$100g \cdot 3$	$50 \cdot 2$	$10 \cdot 2$	13	6
436	$100g \cdot 3$	$50 \cdot 2$	$10 \cdot 3$	6	0

!Hier wird dann die Differenz berechnet, die die kleinste Differenz hat, wird auf der linken Seite platziert,

Differenz= $442-440=2$ und Differenz= $440-439=1$

Also in diesm Fall: 439

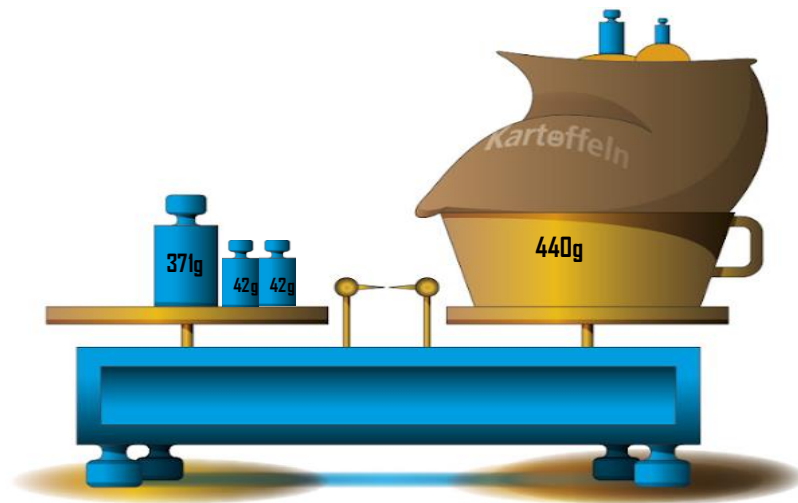
Umsetzung

Wights
<ul style="list-style-type: none"> - lighterW - wights
<ul style="list-style-type: none"> + Wights(lines: String[]) + GetlightW(rechteSeite : int)
Szenario1
<ul style="list-style-type: none"> - linkeSeite
<ul style="list-style-type: none"> + DifferenceCalculator (linkeSeite:int rechteSeite: int Gewichte: ArrayList<Integer>)
Szenario2
<ul style="list-style-type: none"> - linkeSeite - rechteSeite
<ul style="list-style-type: none"> + DifferenceCalculator (linkeSeite:int rechteSeite: int Gewichte: ArrayList<Integer>)
Szenario3
<ul style="list-style-type: none"> - linkeSeite
<ul style="list-style-type: none"> + DifferenceCalculator (linkeSeite:int rechteSeite: int Gewichte: ArrayList<Integer>)
MarktWaage
<ul style="list-style-type: none"> - getWights
<ul style="list-style-type: none"> + BalancbtnBalancbtnActionPerformed (evt: java.awt.event.ActionEvent) +

- ! Die Klasse **Wights** ist eine Liste von Gewichtungen, die Daten aus dem Programm werden gelesen und in der ArrayList (wights) gespeichert, und die leichteren Gewichte werden daraus ausgegeben und in der ArrayList (lighterW) gespeichert.
- Die Klassen **Szenario1,2,3** berechnen die Differenz und speichern am ende das endgewicht in der linkeSeite ,bei der **Szenario2** da wird auch der rechteSeite geändert!
- In der Klasse **MarktWaage** werden die drei Szenarien wie folgt verglichen:
 - Szenario1==0?: Sz1.linkeSeite; //ist der Differenz 0 ?
 - Szenario2== 0?: Sz2.linkeSeite Sz2.rechteSeite; //ist der Differenz 0 ?
 - Szenario3< Szenario1?: Sz3. linkeSeite; // hat der 3te Sz kleinste Differenz?
 - Sonst: Sz1. linkeSeite; //nein? dann hat der 1te Sz das kleinste Differenz

Der Beispiel von der Aufgabe, lässt sich in der 3ten Szenario lösen!!

2000g, 2000g, 2000g, 2000g, 2000g, 371g, 371g, 371g, 371g, 371g, 371g, 127g, 127g, 127g, 42g, 42g, 42g



Kommentar! Ein Gewicht schwerer und am nächsten wäre hier in diesem Beispiel

$$455 = 371g + 42g + 42g$$

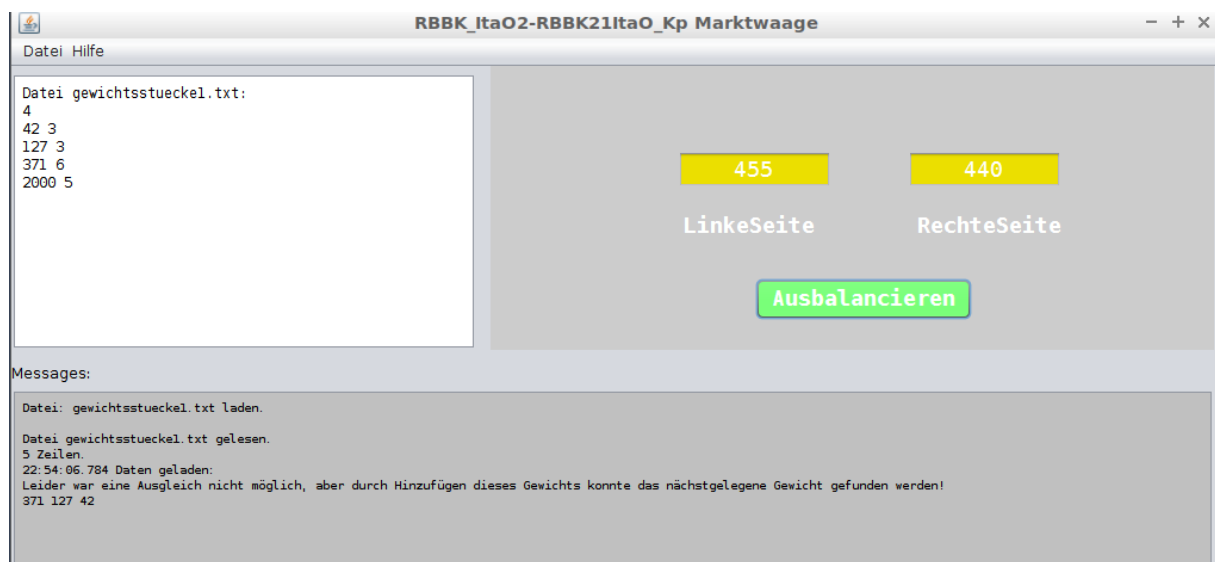
Und ein Gewicht leichter und am nächsten

$$423 = 127g + 127g + 127g + 42g$$

Differenz?

$$455: 455 - 440 = 15 \text{ und } 423: 440 - 423 = 17$$

Also in diesem Fall: die Summe von diesen Gewichte $455 = 371g + 42g + 42g$



Quellcode

```

1
2 import java.util.ArrayList;
3 import java.util.Collections;
4
5 public class Szenario1 {
6
7     static int linkeSeite = 0;
8
9     public int DifferenceCalculator(int linkeSeite, int rechteSeite, ArrayList<Integer> Gewichte) {
10         Collections.sort(Gewichte, Collections.reverseOrder());
11         int temp1 = 0, temp2 = 0;
12
13         for (int i = 0; i < Gewichte.size(); i++) {
14             if (Gewichte.size() == 1) {
15                 this.linkeSeite = Gewichte.get(i);
16                 int differenz = Math.abs(linkeSeite - rechteSeite);
17                 return differenz;
18             }
19             for (int j = i; j < Gewichte.size(); j++) {
20
21                 if (temp1 + Gewichte.get(j) <= rechteSeite) {
22                     temp1 += Gewichte.get(j);
23                     if (temp2 == 0 || temp1 > temp2) {
24                         temp2 = temp1;
25                         linkeSeite = temp2;
26                         if (linkeSeite == rechteSeite) {
27                             break;
28                         }
29                     }
30                 }
31                 if (j == Gewichte.size() - 1) {
32                     temp1 = 0;
33                 }
34             }
35
36         }
37         int differenz = Math.abs(linkeSeite - rechteSeite);
38         this.linkeSeite = linkeSeite;
39         return differenz;
40     }
41 }
42
43
44 import java.util.ArrayList;
45 import java.util.Collections;
46
47 public class Szenario2 {
48
49     static int linkeSeite = 0, rechteSeite = 0;
50
51     int DifferenceCalculator(int linkeSeite, int rechteSeite, ArrayList<Integer> Gewichte) {
52         int differenz = 0;
53         Collections.sort(Gewichte, Collections.reverseOrder());
54         int temp = rechteSeite;
55         for (int i = 0; i < Gewichte.size(); i++) {
56             for (int j = i; j < Gewichte.size(); j++) {
57
58                 if (Gewichte.get(j) > temp && temp > linkeSeite) {
59                     linkeSeite = Gewichte.get(j);
60                 } else if (Gewichte.get(j) + linkeSeite < temp || Gewichte.get(j) + linkeSeite >= temp) {
61                     linkeSeite += Gewichte.get(j);
62                 }
63                 if (linkeSeite > temp) {
64                     for (int k = j; k < Gewichte.size(); k++) {
65                         if (temp + Gewichte.get(k) <= linkeSeite) {
66                             temp += Gewichte.get(k);
67                             Gewichte.remove(k);
68                         }
69                     }
70                     if (temp == linkeSeite) {
71                         rechteSeite = temp;
72                         differenz = Math.abs(linkeSeite - rechteSeite);
73                         this.linkeSeite = linkeSeite;
74                         this.rechteSeite = rechteSeite;
75                         return differenz;
76                     }
77                 }
78             }
79         }
80         if (temp != linkeSeite) {
81             temp = rechteSeite;
82             linkeSeite = 0;
83         }
84
85         differenz = Math.abs(linkeSeite - rechteSeite);
86         this.linkeSeite = linkeSeite;
87         this.rechteSeite = rechteSeite;
88         return differenz;
89     }
90 }

```



```

1
2 import java.util.ArrayList;
3 import java.util.Collections;
4
5 public class Szenario3 {
6
7     static int linkeSeite = 0;
8
9     public int DifferenceCalculator(int linkeSeite, int rechteSeite, ArrayList<Integer> Gewichte) {
10         Collections.sort(Gewichte, Collections.reverseOrder());
11         int temp = 0, temp2 = 0;
12
13         for (int i = 0; i < Gewichte.size() - 1; i++) {
14             for (int j = i; j < Gewichte.size(); j++) {
15                 if (Gewichte.get(j) > rechteSeite) {
16                     linkeSeite = Gewichte.get(j);
17                     j = Gewichte.size();
18                 } else if (temp + Gewichte.get(j) <= rechteSeite) {
19                     temp += Gewichte.get(j);
20                 } else if (temp + Gewichte.get(j) > rechteSeite) {
21                     temp += Gewichte.get(j);
22                     if (temp2 == 0 || temp < temp2) {
23                         temp2 = temp;
24                         linkeSeite = temp2;
25                         temp -= Gewichte.get(j);
26                     } else if (temp >= temp2) {
27                         temp -= Gewichte.get(j);
28                     }
29                 }
30             }
31             if (j == Gewichte.size() - 1) {
32                 temp = 0;
33             }
34         }
35         int differenz = Math.abs(linkeSeite - rechteSeite);
36         this.linkeSeite = linkeSeite;
37         return differenz;
38     }
39 }
40
41

```

Marktwaage.java

```

250 private void BalancbtnActionPerformed(java.awt.event.ActionEvent evt) {
251     int rS = 0;
252     if (!tFRechts.getText().equals("")) || tFRechts.getText() == null) {
253         rS = Integer.parseInt(tFRechts.getText());
254     }
255     Szenario1 s1 = new Szenario1();
256     Szenario2 s2 = new Szenario2();
257     Szenario3 s3 = new Szenario3();
258     getWights().getLightW(rS);
259     if (getWights().lighterW == null || getWights().lighterW.size() < 1) {
260         taMessages.append("\n" + "Bitte Gewichte laden!!");
261     } else if (rS < 1 || rS > 2147483647 || tFRechts.getText().equals("")) {
262         JOptionPane.showMessageDialog(null, "Bitte geben Sie einen gültigen Wert ein!");
263         tFLinks.setBackground(new Color(150, 11, 11));
264         tFRechts.setBackground(new Color(150, 11, 11));
265     } else if (tFRechts.getText().equals(tFLinks.getText())) {
266         taMessages.append("\n" + "Die Entschädigung ist bereits verrechnet!!");
267     } else {
268
269         int V1 = s1.DifferenceCalculator(0, rS, getWights().lighterW);
270         getWights().getLightW(rS);
271         int V2 = s2.DifferenceCalculator(0, rS, getWights().lighterW);
272         getWights().getLightW(rS);
273         int V3 = s3.DifferenceCalculator(0, rS, getWights().lighterW);
274         String linkeSeite = "", rechteSeite = "";
275         if (V1 == 0) {
276             linkeSeite = "" + s1.linkeSeite;
277             tFLinks.setText(linkeSeite);
278             tFLinks.setBackground(Color.GREEN);
279             tFRechts.setBackground(Color.GREEN);
280             taMessages.append("\n" + "Eine Ausgleich war möglich, indem diese Gewichte auf der linken Seite hinzugefügt wurden:\n->" + s1.sum);
281         } else if (V2 == 0) {
282             linkeSeite = "" + s2.linkeSeite;
283             rechteSeite = "" + s2.rechteSeite;
284             tFLinks.setBackground(Color.GREEN);
285             tFRechts.setBackground(Color.GREEN);
286             tFLinks.setText(linkeSeite);
287             tFRechts.setText(rechteSeite);
288             taMessages.append("\n" + "Eine Ausgleich war möglich, indem diese Gewichte auf der linken und Rechten Seite hinzugefügt wurden:\nLinks->" + s2.links + "\nRechts->" + s2.rechts);
289         } else if (V3 < V1) {
290             linkeSeite = "" + s3.linkeSeite;
291             tFLinks.setBackground(new Color(235, 223, 0));
292             tFRechts.setBackground(new Color(235, 223, 0));
293             tFLinks.setText(linkeSeite);
294             taMessages.append("\n" + "Leider war eine Ausgleich nicht möglich, aber durch Hinzufügen dieses Gewichts konnte das nächstgelegene Gewicht gefunden werden!\n" + s3.sum);
295         } else {
296             linkeSeite = "" + s1.linkeSeite;
297             tFLinks.setBackground(new Color(235, 223, 0));
298             tFRechts.setBackground(new Color(235, 223, 0));
299             tFLinks.setText(linkeSeite);
300             taMessages.append("\n" + "Leider war eine Ausgleich nicht möglich, aber durch Hinzufügen dieses Gewichts konnte das nächstgelegene Gewicht gefunden werden!\n" + s1.sum);
301         }
302     }
303 }
304

```