

# Aufgabe 3: Voll Daneben

Team-ID: 00772

Team-Name: T.S

26. November 2018

## Inhaltsverzeichnis

1	Lösungsidee	1
2	Umsetzung	2
3	Beispiele	2
4	Quellcode	2

## 1 Lösungsidee

Für die Auswahl der Zahlen ist es übersichtlicher sich die Glückszahlen der Teilnehmer sortiert auf einem Zahlenstrahl vorzustellen. Die 10 Zahlen können dann folgendermaßen berechnet werden:

1. Man teilt den Bereich von der kleinsten Zahl  $z_1$  bis zur größten Zahl  $z_n$  in 10 gleich lange Abschnitte der Länge  $(z_n - z_1)/10$  (gerundet)
2. Man rechnet alle Zahlen aus je einem Abschnitt zusammen ( $a_k$  für den k-ten Abschnitt) und speichert die Anzahl der Zahlen in dem Abschnitt ( $z_k$  für den k-ten Abschnitt)
3. Für jeden Abschnitt 1,...,10 ergibt sich je eine von Als Zahlen:
  - Ist in dem Abschnitt k mindestens eine Zahl ergibt sich die Zahl  $a_k/z_k$ , also der Durchschnitt
  - Ist der Abschnitt leer, setzt man die obere Grenze des Abschnitts als Zahl

Diese Methode ist zwar nicht optimal (sie könnte optimiert werden indem man die Zahlen der leeren Abschnitte in den nicht leeren Abschnitten benutzt, dazu müsste man aber immer alle Punkte in einem Abschnitt jedes Mal neu speichern und neue Abschnitte berechnen), aber gut dadurch, dass der maximale Abstand zu einem der 10 Zahlen von Beginn an durch die Abschnittslänge begrenzt wird.

Außerdem muss beachtet werden, dass bei weniger als 11 Glückszahlen Als Zahlen einfach die Zahlen selber sein können, sodass die Auszahlung null beträgt.

## 2 Umsetzung

Die Lösungsidee wurde in Java implementiert.

Die Glückszahlen werden in eine Liste eingelesen (Dateiname im Code veränderbar) und sortiert bevor das Verfahren zum Finden von Als Zahlen ausgeführt wird. Nachdem die 10 Zahlen ausgewählt sind, werden die kürzesten Abstände für alle Glückszahlen berechnet, zusammen gerechnet und von dem 25-fachen der Anzahl der Teilnehmer abgezogen um zu sehen, ob Al Verlust oder Gewinn gemacht hat. Die 10 Zahlen und die Bilanz am Ende werden in der Konsole ausgegeben.

## 3 Beispiele

Bei allen gegebenen Beispielen macht Al Gewinn.

### Beispiel 1:

```
1 52, 152, 252, 352, 450, 547, 647, 747, 847, 945,  
  Bilanz: 25
```

### Beispiel 2:

```
55, 158, 244, 354, 432, 549, 666, 763, 845, 930,  
2 Bilanz: 255
```

### Beispiel 3:

```
66, 154, 256, 364, 456, 551, 665, 750, 862, 952,  
2 Bilanz: 75
```

### Beispiel 4 (weniger als 11 Glückszahlen):

```
12, 435, 68, 45, 67, 0, 0, 0, 0, 0, 0,  
2 Bilanz: 125
```

## 4 Quellcode

```
public class VollDaneben {  
2  
    static ArrayList<Integer> gzahlen; //Glueckszahlen der Teilnehmer  
4    static int[] czahlen; // 10 Zahlen  
  
6    public static void main(String[] args) throws FileNotFoundException, IOException {  
        //Einlesen des Textes durch BufferedReader  
8        String filename = "beispiel1.txt";  
        String currentDirectory;  
10       File file = new File(filename);
```

```

currentDirectory = file.getAbsolutePath();
12  final BufferedReader in = new BufferedReader(new FileReader(currentDirectory));
    gzahlen = new ArrayList<>();
14  String line;
    while((line = in.readLine()) != null){
16      gzahlen.add(Integer.parseInt(line));
    }
18  //Fall, dass weniger als 11 Glueckszahlen
    if(gzahlen.size()<11){
20      int bilanz = 25*gzahlen.size();
        for(int j = 0; j<gzahlen.size(); j++){
22          System.out.print(gzahlen.get(j)+"\u0020");
        }
24      for(int k = gzahlen.size(); k<11; k++){
          System.out.print(0+"\u0020");
26      }
        System.out.println();
28      System.out.println("Bilanz:\u0020"+bilanz);
        return;
30    }

32    int subset1 = 0;
    int z1 = 0;
34    ... //Das gleiche fuer 2-9
    Collections.sort(gzahlen);
36    int abschnitt = ((gzahlen.get(gzahlen.size()-1)-gzahlen.get(0))/10);
    int i=0;
38    for(int n = 1; n<=11; n++){
        while(i<gzahlen.size() && gzahlen.get(i)<(gzahlen.get(0)+n*abschnitt)){
40            switch(n){
                case 1:
42                subset1 = subset1 + gzahlen.get(i);
                    z1++;
44                break;
                ... //Das gleiche fuer 2-10
46            }
            i++;
48        }
    }

50    czahlen = new int[10];
    if(z1 != 0) {
52        czahlen[0] = subset1 / z1;
    }else{
54        czahlen[0] = gzahlen.get(0)+ abschnitt;
        ... //Das gleiche fuer 2-10
56        for(int j = 0; j<10; j++){
            System.out.print(czahlen[j)+"\u0020");
58        }
        System.out.println();
60        int verlust = computeMinDistances();
        int gewinn = 25*gzahlen.size();
62        int bilanz = gewinn-verlust;
        System.out.println("Bilanz:\u0020"+bilanz);
64    }

```

```
66      //Berechnet den Betrag den Al auszahlen muss
      static int computeMinDistances(){
68          int distances = 0;
          for(int i = 0; i<gzahlen.size(); i++){
70              distances = distances + getDistance(gzahlen.get(i));
          }
72          return distances;
      }
74
      //findet den Abstand zu der naechsten von Als Zahlen
76      static int getDistance(int gzahl){
          int min = 1000;
78          for(int i = 0; i<10; i++){
              int d = Math.abs(czahlen[i]-gzahl);
80              if(d<min){
                  min = d;
82              }
          }
84          return min;
      }
86 }
```