

# Junioraufgabe 2:

## Vollgeladen

Team-ID: 00370

Team: Linus Schumann

Bearbeiter/-innen dieser Aufgabe:

Linus Schumann

15.11.2021

### Inhaltsverzeichnis

1_Lösungsidee.....	2
2_Umsetzung.....	2
2.1_Allgemeines.....	2
2.2_Laufzeitanalyse .....	2
3_Beispiele .....	2
3.1_Präferenzen 0.....	2
3.2_Präferenzen 1 .....	3
3.3_Präferenzen 2.....	3
3.4_Präferenzen 3.....	4
3.5_Präferenzen 4.....	4
3.6_Präferenzen 5.....	4
4_Quellcode.....	5

# 1\_Lösungsidee

Die Lösungsidee zu dieser Aufgabe besteht darin, dass für jede Person die Daten ausgelesen werden, dann der beste Wert dieser Person berechnet wird. Dann wird der Tag berechnet, an dem die meisten Personen ihren besten Wert haben (allseits beliebter Termin). Da für alle Personen, die an diesem Tag nicht den besten Wert haben, der Wert einfach auf grün gesetzt werden könnte, ist die Anzahl der Wechsel gleich der Anzahl an Personen mit dem besten Wert subtrahiert von der Gesamtanzahl an Personen.

## 2\_Umsetzung

### 2.1\_Allgemeines

Das Programm wurde in Java geschrieben und mit der Java-JDK 17 kompiliert. Im Ordner „executables“ lässt sich die kompilierte .jar-Datei finden und mit der Batchdatei (Windows) oder dem Shell-Script (Linux und MacOS) ausführen. Zusätzlich lassen sich im Ordner „beispieldaten“ alle in dieser Dokumentation erwähnten Beispiele wiederfinden.

Im Ordner „source“ lassen folgende Java Klassen wiederfinden:

- Main: Die Klasse „Main“ ist die Hauptklasse sie übernimmt die Dateiauswahl und gibt den Dateinamen an die andere Klasse weiter
- j2: Die Klasse „j2“ (Junioraufgabe 2) übernimmt die Verarbeitung der Daten und die Ausgabe der Ergebnisse

Wie oben erläutert, wurde das Programm in der Klasse „j2“ umgesetzt. Dies beginnt mit dem Einlesen der Daten in ein zwei-dimensionales Array. Danach wird für jede Person der beste Wert ermittelt und es wird an jedem „bestem Tag“, der Wert (Anzahl an Personen, die diesen Tag als einen „besten Tag“ angegeben haben) des Tages in einem weiteren Array um eins erhöht.

Danach wird der beste Tag ermittelt, indem das Maximum des Arrays ermittelt wird und dieser Tag wird als bester Tag ausgegeben. Zusätzlich wird noch die Differenz der Anzahl der Personen und der Wert dieses Tages aus dem Array ausgegeben (Anzahl an Veränderungen).

### 2.2\_Laufzeitanalyse

Für die Laufzeitanalyse wird einmal die Anzahl an Personen  $n_1$  und die Anzahl an Terminen  $n_2$  betrachtet.

Da für jede Personen, der beste Tag ermittelt wird, beträgt die Laufzeit hierfür  $O(n_1)$ . Für jede Person wird zwei Mal die Liste der Termine durchgegangen. Damit beträgt die Laufzeit dafür  $O(2n_2)$ . Danach wird noch der beste Termin für alle ermittelt, dabei beträgt die Laufzeit  $O(n_2)$ .

Also beträgt die Gesamtlaufzeit für alle Termine und alle Personen  $O(n_1 \cdot 2n_2 + n_2)$ .

## 3\_Beispielen

### 3.1\_Präferenzen 0

	Termin	1	2	3	4	5	6	7	
	1. Person								
	2. Person								
	3. Person								
	4. Person								
	5. Person								
	6. Person								

Ausgabe „praeferenzen0.txt“

Best date: 6

Number of Entries that needs to be changed: 2

### 3.2\_Präferenzen 1

	Termin	1	2	3	4	5	
	1. Person						
	2. Person						
	3. Person						
	4. Person						
	5. Person						

Ausgabe „praeferenzen1.txt“

Best date: 2

Number of Entries that needs to be changed: 1

### 3.3\_Präferenzen 2

	Termin	1	2	3	4	5	6	7	8	9	10
	1. Person										
	2. Person										
	3. Person										
	4. Person										
	5. Person										
	6. Person										
	7. Person										
	8. Person										

*Ausgabe „praeferenzen2.txt“*

Best date: 4

Number of Entries that needs to be changed: 0

**3.4\_Präferenzen 3**

Termin	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1. Person																				
2. Person																				
3. Person																				
4. Person																				
5. Person																				
6. Person																				
7. Person																				
8. Person																				
9. Person																				
10. Person																				
11. Person																				
12. Person																				
13. Person																				
14. Person																				

*Ausgabe „praeferenzen3.txt“*

Best date: 18

Number of Entries that needs to be changed: 7

**3.5\_Präferenzen 4***Ausgabe „praeferenzen4.txt“*

Best date: 22

Number of Entries that needs to be changed: 14

**3.6\_Präferenzen 5***Ausgabe „praeferenzen5.txt“*

Best date: 31

Number of Entries that needs to be changed: 34

## 4\_Quellcode

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class j2 {
    public static void calculateChanges(String filename) throws
FileNotFoundException{
        //__READ-FILE__//
        File file = new File(filename);
        Scanner scanner = new Scanner(file);
        int numberOfMembers = scanner.nextInt();
        int numberOfDates = scanner.nextInt();
        //__Save data into 2D-Array__//
        int[][] dataSet = new int[numberOfMembers][numberOfDates];
        for(int i = 0; i < numberOfMembers; i++){
            for(int j = 0; j < numberOfDates; j++){
                dataSet[i][j] = scanner.nextInt();
            }
        }
        //__Calculate bestAnswers 2D-Array__//
        int[] bestAnswers = new int[numberOfDates];
        for(int i = 0; i < numberOfMembers;i++){
            //__Calculate the best Answer of one Member__//
            int bestAnswer = 2;
            for(int j = 0; j < numberOfDates;j++){
                if(dataSet[i][j] < bestAnswer){
                    bestAnswer = dataSet[i][j];
                }
            }
            //__For each day add best Answer__//
            for(int j = 0; j < numberOfDates; j++){
                if(dataSet[i][j] == bestAnswer){
                    bestAnswers[j] += 1;
                }
            }
        }
        //__Calculate best Date__//
        int bestDate = 0;
        int bestDateNumber = 0;
        for(int i = 0; i < numberOfDates; i++){
            if(bestAnswers[i] > bestDateNumber){
                bestDateNumber = bestAnswers[i];
                bestDate = i+1;
            }
        }
        System.out.println("Best date: "+bestDate);
        System.out.println("Number of Entries that needs to be changed:
" +(numberOfMembers-bestDateNumber));
    }
}
```