Day 1

Aufgabe und Input

- Aufgabe
- Input

(d1e - Beispieleingabe aus dem Aufgabentext, d1i Eingabe für die Lösungen auf dieser Wikiseite)

Ergebnisse

Ergebnis Teil 1 für die Eingabe auf dieser Wikiseite (weicht ab von deiner "echten" Lösung)

70509

Ergebnis Teil 2 für die Eingabe auf dieser Wikiseite (weicht ab von deiner "echten" Lösung)

208567

Hilfestellungen

Tipp 1

Prinzipiell muss man die Eingabedatei Zeile für Zeile lesen und die Zahlen solange addieren, bis eine Leerzeile kommt. Nun kann man entscheiden, ob die so erhaltene Summe die größte ist - wenn ja, ersetzt man die "bisherige größte Summe durch diese neue größte Summe.

Tipp 2: Kommentiertes Programmgerüst für Teil 1

```
public int partOne() {
        int maxCalories=0;
        int packCalories=0;
        for(String line[]: input) {
            String zeilenString=line[0];

            if(!zeilenString.equals("")) {
                // Wenn es keine Leerzeile ist:
                // zu packCalories hinzuzählen, dabei
                // muss der String in eine Integer Zahl umgewandelt werden
                // mit Integer.parseInt();
```

```
                FIXME Hier fehlt was
            } else {
                // Wenn es eine Leerzeile ist:
                // Schauen, ob packCalories größer ist als maxCalories.
                // Wenn ja, merken, dann packCalories auf Null
                // zurücksetzen.
                FIXME Hier fehlt was
            }
        }

        return maxCalories;
    }
```

Tipp 3: Lösungsvorschlag für Teil 1

```
public int partOne() {
        int maxCalories=0;
        int packCalories=0;
        for(String line[]: input) {
            String zeilenString=line[0];

            if(!zeilenString.equals("")) {
                // Wenn es keine Leerzeile ist:
                // zu packCalories hinzuzählen, dabei
                // muss der String in eine Integer Zahl umgewandelt werden
                // mit Integer.parseInt();
                packCalories=packCalories+Integer.parseInt(zeilenString);
            } else {
                // Wenn es eine Leerzeile ist:
                // Schauen, ob packCalories größer ist als maxCalories.
                // Wenn ja, merken, dann packCalories auf Null
                // zurücksetzen.
                if (packCalories > maxCalories) {
                    maxCalories = packCalories;
                }
                packCalories=0;
            }
        }

        return maxCalories;
    }
```

Tipp 4: Ein kleiner Hinweis zu Teil 2

Für den Teil zwei muss man am Ende die 3 größten Kalorienpakete kennen, also muss man prüfen, wo sich das aktuelle Paket in dieser "Rangliste der ersten drei einordnet" und dieses entsprechend anpassen.

Alternativ kann man *alle* Kalorienpakete in eine ArrayList speichern, diese am Ende sortoieren und dann die ersten drei Elemente addieren.

From:
https://info-bw.de/ -

Permanent link:
https://info-bw.de/faecher:informatik:oberstufe:java:aoc:aoc2022:day1:start?rev=1669910355

Last update: **01.12.2022 15:59**