Tag 4

- Variante 1

Hilfestellungen Teil 1

- Grundidee: Nutze die Methode `split("…")`, um die Zeile so zu unterteilen, dass du zunächst alle Zahlen vor unter hinter dem Trennzeichen | bekommst. Nutze es erneut, um vom vorderen Teil wiederum nur die Zahlen hinter dem Doppelpunkt zu bekommen. Splitte die beiden Strings mit Zahlen erneut an den Leerzeichen, um alle einzelnen Zahlen als Stringarray zu bekommen, diese kannst du dann der Reihe nach mit `Integer.parse(…)` in int-Zahlen umwandeln. Anschließend überprüfst du für jede hintere Zahl, ob sie in den vorderen Zahlen vorkommt.
- Bei zwei Verwendungen von `split` musst du aufpassen! **Erstens**: Der senkrechte Strich wird im Java-String als Metazeichen erkannt und muss "escaped" werden (also die versteckte Funktion/Bedeutung muss entfernt werden) `\\|`. **Zweitens**: zwischen manchen Zahlen sind mehrere Leerzeichen. Du musst also machnmal mehrere Leerzeichen zum Splitten nutzen. Das kannst du mit dem "regulären Ausdruck" `\\s+` machen, dieser erkennt beliebig viele "Whitespaces" (also u.a. das Leerzeichen).
- Zähle, wie häufig die hinteren Zahlen in den vorderen Zahlen vorkamen. Die Punkte pro Zeile berechnest du dann mit 2^(anzahl-1)

Lösungsvorschlag 1

```
private int[] stringToIntArray(String str) {
    String[] numbersStr = str.trim().split("\\s+");
    int[] array = new int[numbersStr.length];
    for (int i = 0; i < numbersStr.length; i++) {
        array[i] = Integer.parseInt(numbersStr[i].trim());
    }
    return array;
}

public int partOne() {
    int summe = 0;

    for (String line: inputLines) {
        String winningNumbersStr, numbersYouHaveStr;
        winningNumbersStr = line.split("\\|")[0].trim().split(":")[1];
        numbersYouHaveStr = line.split("\\|")[1].trim();

        int[] winningNumbers, numbersYouHave;
        winningNumbers = stringToIntArray(winningNumbersStr);
        numbersYouHave = stringToIntArray(numbersYouHaveStr);

        int matches = 0;
        for (int number : numbersYouHave) {
            for (int winningNumber : winningNumbers) {
                if (number == winningNumber) {
                    matches++;
```

```
                    }
                }
            }
            summe += Math.pow(2, matches-1);
        }

        return summe;
    }
```

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

Permanent link:
https://info-bw.de/faecher:informatik:oberstufe:java:aoc:aco2023:day4:start?rev=1701681763

Last update: **04.12.2023 09:22**