

- [Variante 1](#)

Hilfestellung Teil 1

Die Aufgabe ist schwierig zu beschreiben und entsprechend sind die hier gelisteten Tipps möglicherweise nur eingeschränkt hilfreich.

- Es ist sinnvoll, die Karten nach zwei Schemas zu sortieren. Einmal nach der Kartentyp (5 identisch, 4 identisch, etc.) und dann pro Typ noch nach dem Wert der einzelnen Karten (A, K, Q, ...).
- Ein möglicher Datentyp, um all diese Dinge zu berücksichtigen, ist eine `ArrayList<ArrayList<String[]>>`. Was ist damit gemeint? Die äußere `ArrayList` enthält insgesamt 7 `ArrayLists`, welche jeweils alle Karten nach Kartentyp sortiert enthalten. Die erste innere `ArrayList` enthält also z. B. alle Karten, bei denen alle 5 identisch sind, etc. Pro innerer `ArrayList` wird ein `String-Array` gespeichert. Dieses stellt die beiden `String`-Bestandteile jeder Zeile dar. Dadurch ist gewährleistet, dass der Kartenwert jeder Karte immer fest zugeordnet ist und auffindbar bleibt (auch nach Umsortierungen).
- Die inneren `String-Arrays` können als erstes erstellt werden.
- Pro `String-Array` muss nun der Typ der Karte ermittelt werden und das `String-Array` der passenden `ArrayList` zugefügt werden.
- Am Ende muss man alle eingetragenen `ArrayLists` in der richtigen Reihenfolge durchlaufen, die `String-Arrays` pro `ArrayList` korrekt sortieren und alle Werte zusammenrechnen.

From:

<https://www.info-bw.de/> -

Permanent link:

<https://www.info-bw.de/faecher:informatik:oberstufe:java:aoc:aco2023:day7:start?rev=1701957659>

Last update: **07.12.2023 14:00**

