Tag 15 - Lens Library

- Variante 1

Tag 15 ist meines Erachtens der bisher einfachste Tag. Insbesondere Teil 1 war sehr einfach! Teil 2 war umfangreicher, aber auch da gab es keine besonders großen Hürden.

Lösungshinweise Teil 1

- Die Eingabedatei enthält dankenswerterweise nur eine einzelne Zeile. Diese muss bei den Kommas mit `split(",")` in ein String-Array aufgesplittet werden.
- Anschließend iteriert man über jeden String und berechnet stumpfsinnig der Aufgabenstellung folgend die Hashes die man auf die Gesamtsumme aufsummiert. Denke dabei an den Modulo-Operator %.

Lösungsvorschlag Teil 1

```
public int partOne() {
    int summe = 0;

    String[] sequences = inputLines.get(0).split(",");
    for (String s: sequences) {
        int hash = 0;
        for (char c: s.toCharArray()) {
            hash += (int)c;
            hash *= 17;
            hash = hash % 256;
        }
        summe += hash;
    }

    return summe;
}
```

Lösungshinweise Teil 2

Für Teil 2 muss man sich Gedanken über die Datenstruktur machen. Man hat insgesamt 256 Boxen, das spricht für eine ArrayList, die flexiblere Datentypen enthalten kann als ein Array. Pro Box hat man wiederum mehrere Linsen. Also wiederum eine ArrayList pro Box. Eine Linse wiederum muss sowohl einen String (label) als auch einen int (focalLength) speichern, das spricht für eine eigene Klasse für Linsen. Das ergibt insgesamt die Datenstruktur: `ArrayList<ArrayList<Lens>`.

- Fange damit an, die `Lens`-Klasse zu erstellen. Sie muss einfach nur einen String und einen int als Instanzvariablen enthalten sowie die nötigen getter- und setter-Methoden.

Lösungsvorschlag Lens-Klasse

```
public class Lens
```

```
{
    private String label;
    private int focalLength;

    /**
     * Konstruktor für Objekte der Klasse Lens
     */
    public Lens(String label, int focalLength)
    {
        this.label = label;
        this.focalLength = focalLength;
    }

    public void setLabel(String label) {
        this.label = label;
    }

    public String getLabel() {
        return this.label;
    }

    public void setFocalLength(int focalLength) {
        this.focalLength = focalLength;
    }

    public int getFocalLength() {
        return this.focalLength;
    }
}
```

- Erstelle die oben gesprochene große Datenstruktur für die 256 Boxen. Stelle mit einer Schleife sicher, dass 256 innere `ArrayList<Lens>` existieren.
- Gehe nun in einer Schleife über jede Sequenz:
  - Teile die Sequenz, um das Label zu bekommen. Tipp: nutze den regulären Ausdruck `"[=\\-]"` in der `split()`-Methode, um entweder an `-` oder an `=` zu splitten.
  - Berechne für das Label den Hash.
  - "Hole dir" die korrekte Box in Abhängigkeit des Hash-Ergebnisses
  - Nun musst du pro Box unterscheiden, ob in der Sequenz ein `=` oder ein `-` enthalten war.
    - Bei einem Gleichheitszeichen musst du zunächst prüfen, ob das Label bereits in der Box enthalten ist. (Prüfe für jede Lens innerhalb der aktuellen Box, ob das Label passt). Gibt es einen Treffer, dann aktualisierst du die focalLength (diese musst du durch einen weiteren Split der aktuellen Sequenz entnehmen). Gibt es keinen Treffer, dann fügst du ein neues Lens-Objekt der Box hinzu.
    - Andernfalls beim Minuszeichen musst du die Box durchsuchen bis du das Label gefunden hast und musst dann die Box an diesem Index löschen.
- Am Ende gehst du über alle Boxen und Linsen drüber und summierst die Werte auf.

Lösungsvorschlag Teil 2

```
public long partTwo() {
    int summe = 0;

    ArrayList<ArrayList<Lens>> boxes = new ArrayList<ArrayList<Lens>>();
    for (int i = 0; i < 256; i++) {
        boxes.add(new ArrayList<Lens>());
    }

    String[] sequences = inputLines.get(0).split(",");
    for (String s: sequences) {
        String label = s.split("[=\\-]")[0];

        // berechne den Hash des labels
        int hash = 0;
        for (char c: label.toCharArray()) {
            hash += (int)c;
            hash *= 17;
            hash = hash % 256;
        }

        // Nimm die richtige box
        ArrayList<Lens> box = boxes.get(hash);

        // entweder eine lens hinzufügen oder aktualisieren
        if (s.contains("=")) {
            int focalLength = Integer.parseInt(s.split("[=\\-]")[1]);
            boolean alreadyExists = false;
            for (Lens l: box) {
                if (l.getLabel().equals(label)) {
                    alreadyExists = true;
                    l.setFocalLength(focalLength);
                }
            }
            if (!alreadyExists) {
                box.add(new Lens(label, focalLength));
            }
        }
        // oder die lens löschen
        else {
            for (int i = 0; i < box.size(); i++) {
                if (box.get(i).getLabel().equals(label)) {
                    box.remove(i);
                }
            }
        }
    }

    // alle Werte zusammenrechnen
    for (int b = 0; b < boxes.size(); b++) {
        ArrayList<Lens> box = boxes.get(b);
        for (int s = 0; s < box.size(); s++) {
```

```
                summe += (b + 1) * (s + 1) * box.get(s).getFocalLength();
            }
        }

        return summe;
}
```