Day 4: Camp Cleanup

Aufgabe und Input

- Aufgabe
- Input

(d4e - Beispieleingabe aus dem Aufgabentext, d4i Eingabe für die Lösungen auf dieser Wikiseite)

Ergebnisse

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

466

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

865

Tipps und Hinweise

Methodengerüst Basis

```
    public day4() throws Exception {
        // Einlesen am Komma getrennt, wir erhalten zwei Felder.
        // line[0] - erster Bereich
        // line[1] - zweiter Bereich
        this.readInput(inputFile,',');
        this.printInput();
    }

    public int partOne() {
        int answer=0;
        for(String[] line: input) {
            // Bereiche trennen
            String[] bereich1 = line[0].split("-");
            String[] bereich2 = line[1].split("-");
            // Kontrollausgabe
            System.out.println("[Bereich 1] " + bereich1[0] + " bis " +
bereich1[1]);
            System.out.println("[Bereich 2] " + bereich2[0] + " bis " +
bereich2[1]);
```

```
            // Grenzen mit "sprechenden Namen" bezeichnen
            int start1 = Integer.parseInt(bereich1[0]);
            int start2 = Integer.parseInt(bereich2[0]);
            int end1 = Integer.parseInt(bereich1[1]);
            int end2 = Integer.parseInt(bereich2[1]);

            boolean overlapping=false;

            // Hier müssen Bedingungen (if ...) eingefügt werden, die
            // überprüfen, ob sich die Bereiche überlappen.
            // Wenn ja, setzt mal overlapping = true
            // dann werden die Bereiche anschließend als überlappend
            // zur Antwort addiert

            if (overlapping) {
                answer++;
            }

        }

        System.out.println(answer);
        return answer;
    }
```

Methodengerüst mit Beispielbedingung

```
    public day4() throws Exception {
        // Einlesen am Komma getrennt, wir erhalten zwei Felder.
        // line[0] - erster Bereich
        // line[1] - zweiter Bereich
        this.readInput(inputFile,',');
        this.printInput();
    }

    public int partOne() {
        int answer=0;
        for(String[] line: input) {
            // Bereiche trennen
            String[] bereich1 = line[0].split("-");
            String[] bereich2 = line[1].split("-");
            // Kontrollausgabe
            System.out.println("[Bereich 1] " + bereich1[0] + " bis " +
bereich1[1]);
            System.out.println("[Bereich 2] " + bereich2[0] + " bis " +
bereich2[1]);
            // Grenzen mit "sprechenden Namen" bezeichnen
            int start1 = Integer.parseInt(bereich1[0]);
            int start2 = Integer.parseInt(bereich2[0]);
```

```
            int end1 = Integer.parseInt(bereich1[1]);
            int end2 = Integer.parseInt(bereich2[1]);

            boolean overlapping=false;

            // Erste Bedingung
            if (start1 < start2 && start2 <= end1) {
                // Bereich 2 kann in Bereich 1 enthalten sein, ist es aber
nur dann
                // wenn end2 <= end1 ist
                if (end2 <= end1) {
                    System.out.println("Bereich 2 ist in Bereich 1
enthalten");
                    overlapping=true;
                }

            }

            // Hier müssen weitere Bedingungen (if ...) eingefügt werden,
die
            // andere Möglichkeiten überprüfen, wie sich die Bereiche
überlappen
            // können.


            if (overlapping) {
                answer++;
            }

        }

        System.out.println(answer);
        return answer;
    }
```