

# Coding Challenge Lösung, Leitfaden und Diagramme

Charissa Morales

März 2024

## 1 Endgültige Lösung

### 1.1 Diagramm

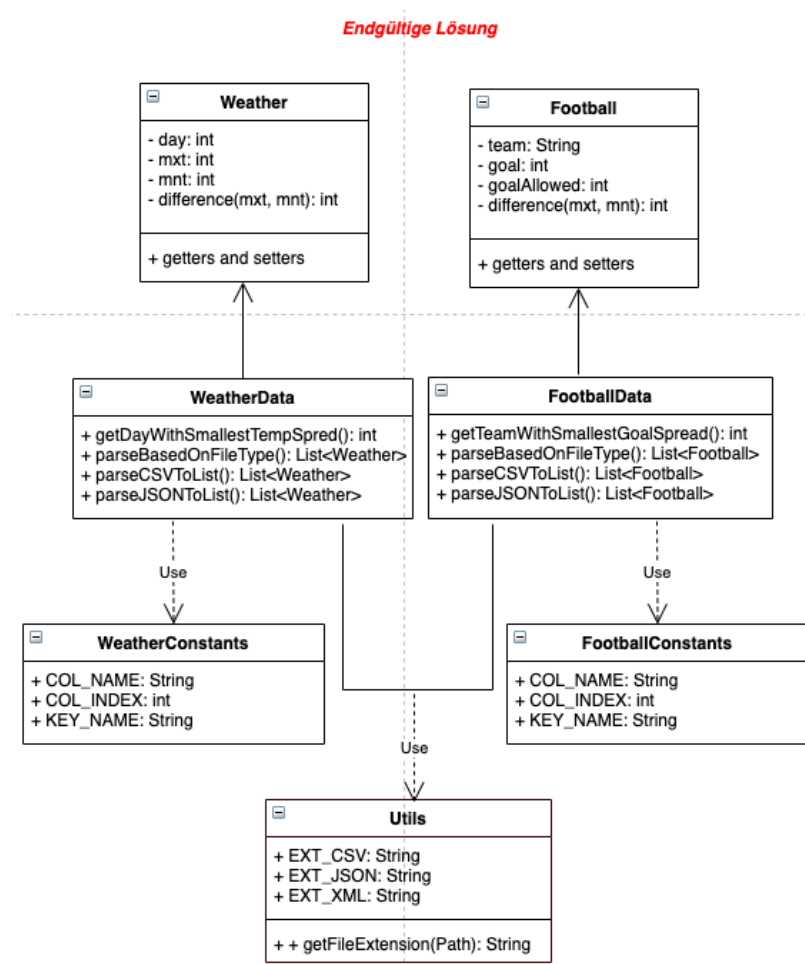


Figure 1: *nicht alle Variablen sind im Diagramm dargestellt*

## 1.2 Pseudo-Code

1. Datei lesen (where 1 Zeile = 1 Objekt)
2. Eine Liste von Objekten erstellen:  

$$\text{List<Object> ( where 1 List<Object> = 1 File )}$$
3. Differenz(var1,var2) berechnen:
  - Weather — var1: maximum temperature, var2: minimum temperature
  - Football — var1: goals, var2: goals allowed
4. Sortieren der Objektliste in aufsteigender Reihenfolge nach der Differenz
5. Abfragen der Daten im ersten Index
6. Getters verwenden, um Day (für Weather) und Team (für Football) zu erhalten.

## 2 Sonstige Ansätze (*Ansätze, die ich nicht geeignet finde*)

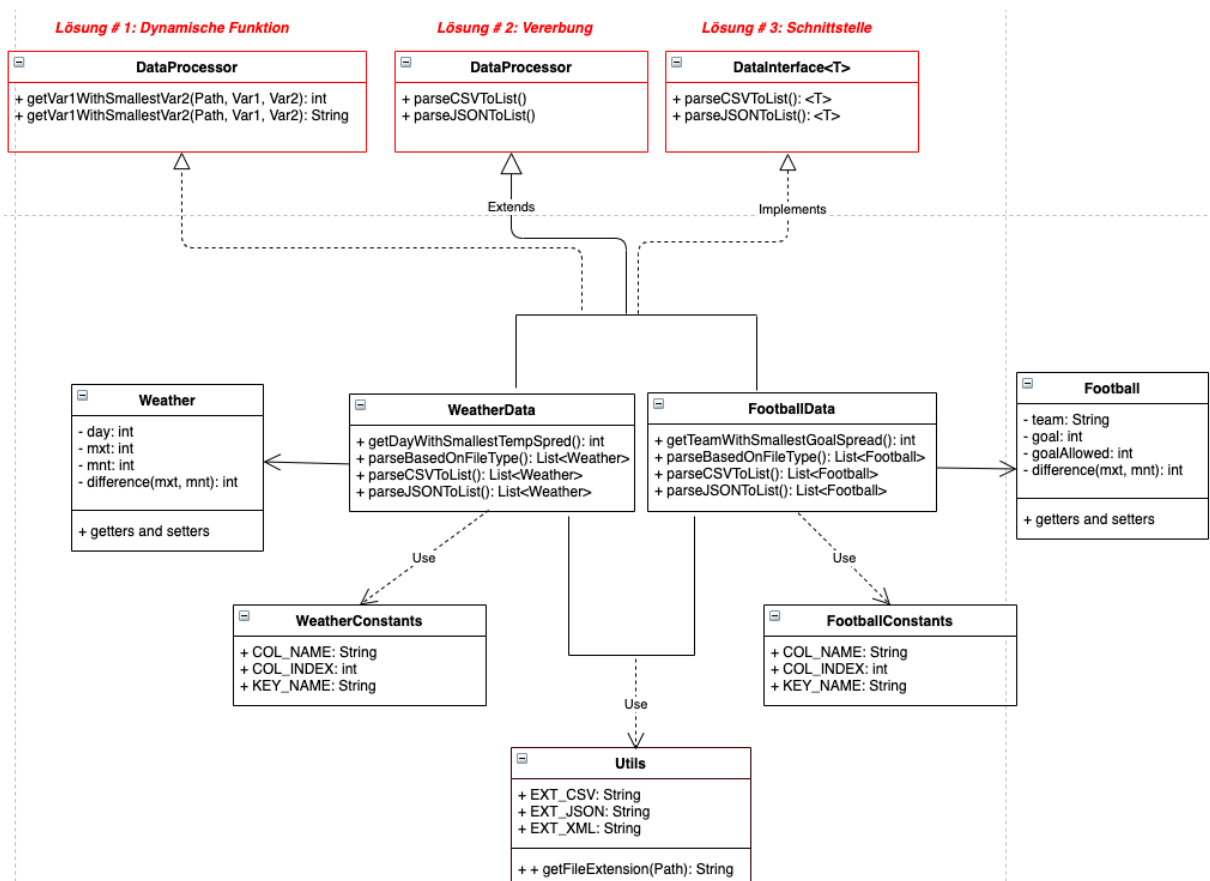


Figure 2: *nicht alle Variablen sind im Diagramm dargestellt*

1. Dynamic Function (*Dynamisches Funktion*)

Lösung: Nicht Geeignet

Grund: Die erste Schnittstelle ist nur für `WeatherData` Klasse geeignet und die zweite Schnittstelle ist für `FootballData` geeignet.

2. Inheritance (*Vererbung*)

Lösung: Nicht Geeignet

Grund: Die Objektattribute, die von den Parsing-Funktionen verarbeitet werden, unterscheiden sich je nach Objekttyp (`Weather` und `Football`)

3. Interface (*Schnittstelle*)

Lösung: Nicht Geeignet

Grund: Die Methoden `parseCSVToList()` und `parseJSONToList()` werden lokal in den Klassen `WeatherData` und `FootballData` verwendet. Es wäre sinnvoll, diesen Ansatz zu verwenden, wenn die Parse-Funktionen außerhalb dieser Klassen verwendet werden.