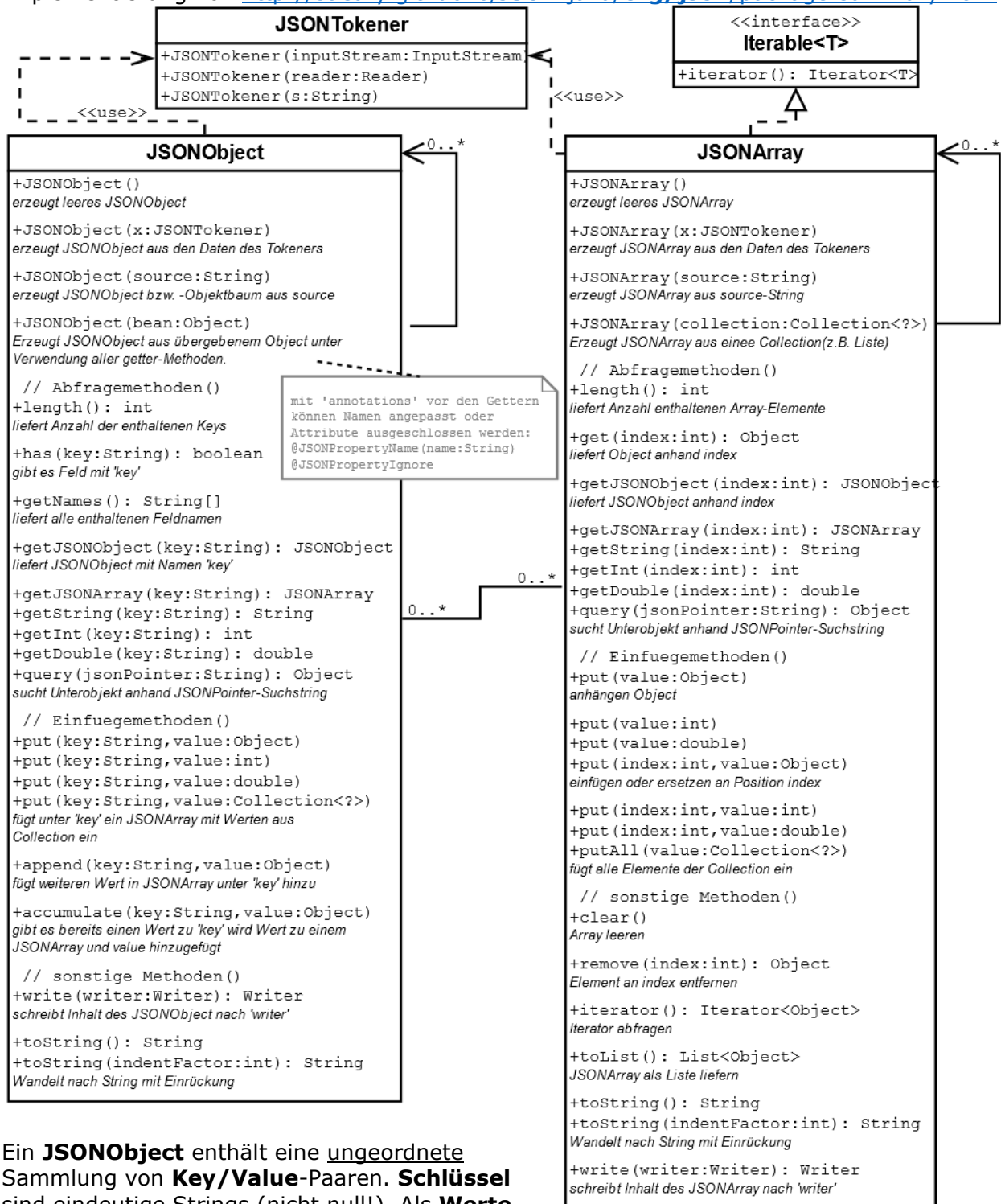


UML-Klassendiagramm zeigt einen Teilausschnitt der zentralen Klassen der Java-JSON-Implementierung von <http://stleary.github.io/JSON-java/org/json/package-summary.html>

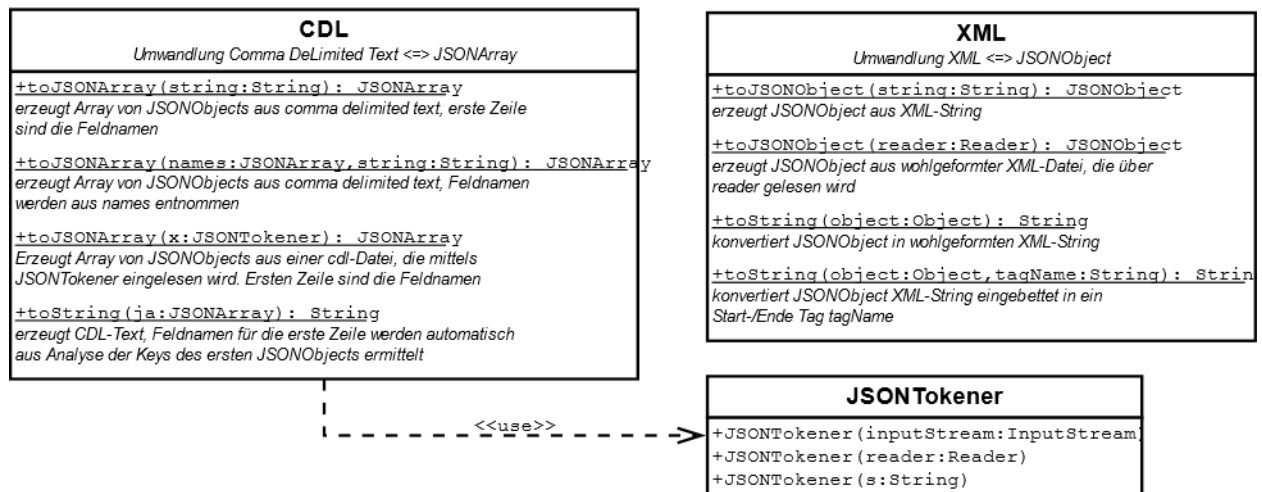


Ein **JSONObject** enthält eine ungeordnete Sammlung von **Key/Value**-Paaren. **Schlüssel** sind eindeutige Strings (nicht null!). Als **Werte** sind Wahrheitswerte, Zahlen, Strings, JSONArray-Objekte oder wiederum JSONObject-Objekte zulässig.

Ein **JSONArray** ist eine geordnete Sammlung von **Werten** (Wahrheitswerte, Zahlen, Strings, JSONObject-Objekte, oder auch JSONArrays). Der Zugriff erfolgt über einen **Index**.

Die **get(...)-Methoden** beider Klassen lösen bei ungültigen Schlüssel/Indizes eine **JSONException** aus. Alternativ gibt es auch **opt(...)-Methoden** (im Diagramm nicht gezeigt) die **bei Fehlern null** zurückliefern.

Weitere Hilfsklassen mit statischen Methoden zur Konvertierung zwischen **JSON** (-Objekten /-Arrays) und „Comma **De**Limited Text“ (CSV) bzw. **XML**:



Zur Nutzung der „org.json – Klassen“ muss das zugehörige JAR-Archiv (json-20230618.jar oder neuer) in das eclipse Projekt hinzugefügt werden (Build Path > Add External Archives) oder alternativ das Repository per maven, ... eingebunden werden.

Anwendungsbeispiele der Bibliothek sind zu finden unter:

<https://www.baeldung.com/java-org-json>

<https://zetcode.com/java/json/>