



ulm university universität
uulm

**Fakultät für
Ingenieurwissenschaften,
Informatik und
Psychologie**

Institut für Software-
technik und Program-
miersprachen

Softwaregrundprojekt Meilenstein 6

Softwaregrundprojekt an der Universität Ulm

Vorgelegt von:

Gruppe 10

Dozent:

Florian Ege

Betreuer:

Stefanos Mytilineos

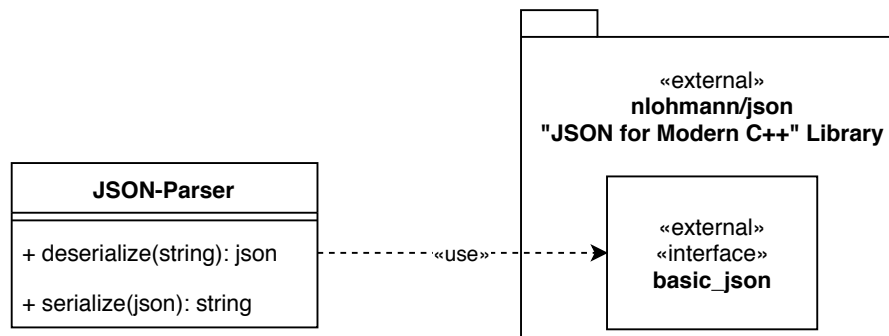
2019

Inhaltsverzeichnis

1	JSON-Parser	3
1.1	Klassen-Diagramm	3
1.2	Beschreibung	3
1.2.1	JSON-Parser (Klasse)	3
1.2.2	basic_json (Externe Bibliothek)	3
1.3	Zuordnung der Funktionalen Anforderungen	4

1 JSON-Parser

1.1 Klassen-Diagramm



1.2 Beschreibung

1.2.1 JSON-Parser (Klasse)

deserialize (Methode) Die deserialize Methode dient dazu einen dem JSON-Format konformen String in ein JSON-Object umzuwandeln. Dabei wird bei der Umsetzung dieser Funktionalität auf Methoden und Strukturen aus einer Software-Bibliothek eines Drittanbieters zurückgegriffen.

serialize (Methode) Die serialize Methode dient dazu aus einem JSON-Object ein dem JSON-Format genügenden String zu erzeugen. Wie auch schon bei der deserialize Methode wird dabei auf Methoden und Strukturen aus einer Software-Bibliothek eines Drittanbieters zurückgegriffen.

1.2.2 basic_json (Externe Bibliothek)

Bei dieser Klasse handelt es sich um eine externe Bibliothek, die verschiedenste Methoden und Strukturen zur Verfügung stellt um in einem C++ Projekt mit Objekten mit JSON-Struktur einfach zu arbeiten. Vermutlich wird die von *Niels Lohmann* entwickelte freie JSON Bibliothek namens *JSON for Modern C++* verwendet werden. Genauere Informationen zu dieser Bibliothek sind unter folgendem Link zu finden:

<https://github.com/nlohmann/json>

1.3 Zuordnung der Funktionalen Anforderungen

Die funktionalen Anforderungen gemäß dem Pflichtenheft werden den Komponenten folgendermaßen zugeteilt:

Methode	Abgedeckte funktionale Anforderungen
deserialize	keine explizite FA vorhanden
serialize	keine explizite FA vorhanden