

# Compilerbau

## – Prüfungsleistung –

Dozent: Martin Plümicke , Andreas Stadelmeier

WS 2024/25

### Spezifikation

**Deklarationen:** •  $\Sigma$ : Eingabe-Alphabet

- JC: Menge aller syntaktisch korrekten Java-Klassen der Untermenge von Java, die in der Vorlesung vorgestellt wurde.
- BC: Menge aller Bytecode-Files

**Eingabe:**  $p \in \Sigma^*$

**Vorbedingung:**  $\emptyset$

**Ausgabe:**  $bc \in BC^* \cup \{error\}$

- Nachbedingungen:**
- falls  $p \in (JC)^*$ , so ist  $bc \in (BC)^*$  und  $p$  wird nach  $bc$  übersetzt wie es durch die Sprache Java definiert ist.
  - falls  $p \notin (JC)^*$ , so ist  $bc = error$ .

### Vorgehen

**Arbeitsteam:** Der Java-Compiler wird in einem Team von 3 Personen erstellt. Das Team wird nochmals unterteilt:

- **Scannen/Parsen (1 Personen)**

**Abstrakte Syntax:** Aufbau der abstrakten Syntax aus dem Parsetree (Output von ANTLR), (Es kann die in Netz verfügbare ANTLR-Grammatik genutzt werden)

- **Semantische Analyse (1 Person)**

**Typisierung:** Typisierung der abstrakten Syntax

- **Codeerzeugung (1 Person):** Erzeugung des Bytecodes mit ASM

## Prüfungsleistung

Die Arbeitsleistung einer/s jeden Studierenden wird bewertet an Hand folgender Kriterien:

- Gesamtes Projektergebnis (20%)
- Eigene Leistung(60%)
- Mitarbeit im Team (20%)

Das Projektergebnis muss folgendes beinhalten:

- Kurzdokumentation:
  - Bedienungsanleitung des Programms
  - UML-Klassendiagramm der abstrakten Syntax
  - Dokumentation wer welche Aufgaben erledigt hat.
- (Git-)Repository mit dem Java-Code
- Eine Testsuite von Java-Programmen, für die der Compiler funktioniert.
- **Abgabe- und Präsentationstermin: Letzte Vorlesungswoche**