

# Compilerbau

## – Prüfungsleistung –

Dozent: Martin Plümicke

WS 2021/22

### Spezifikation

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

- JC: Menge aller syntaktisch korrekten Java-Klassen mit folgenden Einschränkungen:
  - keine generischen Klassen
  - keine abstrakten Klassen
  - keine Vererbung
  - keine Interfaces
  - keine Threads
  - keine Exceptions
  - keine Arrays
  - als Basistypen sind nur `int`, `boolean` und `char` zugelassen
  - keine Packages
  - keine Imports
  - keine Lambda-Expressions
- 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

**Arbeitssteam:** Der Java-Compiler wird in einem Team von 4-5 Personen erstellt. Das Team wird nochmals unterteilt:

- **Scannen/Parsen/Abstrakte Syntax/Projektsteuerung (2 Personen)**
  - ANTLR-File:** Erstellen des G4-Files oder
  - JLEX/jay-File:** Erstellen eines JLex- und Jay-Files
  - Abstrakte Syntax:** Aufbau der abstrakten Syntax
- **Semantische Analyse (1 Person)**
  - Typisierung:** Typisierung der abstrakten Syntax
- **Codeerzeugung (1 Person):** Erzeugung des Bytecodes mit ASM
- **Tester (1 Person)**
  - Testsuite von Java-Files, die alle implementierten Features abdecken.
  - Händische Übersetzung aller Java-Files der Testsuite in die abstrakte Syntax (als Test-Eingaben für den Typ-Checker)
  - JUnit-Tests für Parser und Erzeugung der abstrakten Syntax
  - Händische Typisierung aller Testfälle der abstrakten Syntax (als Eingabe für die Codeerzeugung)
  - JUnit-Tests für die Typisierung

## Prüfungsleistung

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

- Projektergebnis
- wöchentlicher Projektfortschritt
- Mitarbeit im Team

Das Projektergebnis muss folgendes beinhalten:

- Kurzdokumentation aus der hervorgeht welche Leistung der jeweiligen Studierende erbracht hat.
- Im Teilprojekt muss folgendes vorliegen:
  - Eine Testsuite von Java-Programmen, für die der Compilerteil funktioniert.
  - Präsentation des Programms an Hand der erstellten Testsuites.
- Durchgehendes Beispiel, für das der gesamte Compiler funktioniert.
- **Abgabetermin: Letzter Tag des Semesters**