

Compilerbau

– Prüfungsleistung –

Dozent: Martin Plümicke, Andeeas Stadelmeier

WS 2024/25

Spezifikation

Deklarationen: • Σ : 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 4 (–5) Personen erstellt. Das Team wird nochmals unterteilt:

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

Scannen: alex-File oder Scanner von Hand programmieren

Grammatik: Erstellen einer Mini-Java-Grammatik an Hand der Spezifikation

Parsen: Erstellen des happy-Files oder des Kombinator-Parsers und Aufbau des abstrakten Syntaxbaums

- **Semantische Analyse (1 Person) Typisierung der abstrakten Syntax (1 Person)**

- **Codeerzeugung (2–3 Personen):**

- Aufbau eines abstrakten ClassFiles (1 Person)

- Konstantenpool (1 Person)

- **Nur bei Bearbeitung durch 3 Personen:** Umwandlung des ClassFiles in Bytecode (1 Person)

Prüfungsleistung

Die Arbeitsleistung wird bewertet an Hand

- des Gesamtergebnis des Teams
- der Arbeitsleistung jeder/s Studierenden

an Hand folgender Kriterien:

- Projektergebnis
- wöchentlicher Projektfortschritt
- Mitarbeit im Team

Das Projektergebnis muss folgendes beinhalten:

- (Kurz)dokumentation 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 erstellen Testsuites.
- Durchgehendes Beispiel, für das der gesamte Compiler funktioniert.
- **Abgabetermin: Letzte Semesterwoche**