Compilerbau
– Prüfungsleistung –

Dozent: Martin Plümicke

WS 2018/19

Spezifikation

Deklarationen:
- Σ: 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: $bc \in \mathsf{BC}^* \cup \{error\}$

Nachbedingungen:
- falls $p \in (JC)^*$, so ist $bc \in (\mathsf{BC})^*$ 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 8 Personen erstellt. Das Team wird nochmals unterteilt:

- **Scannen/Parsen/Projektleitung (1 Person)**

  Scannen: `alex`–File programmieren

  Parsen: Erstellen des `happy`–Files und (Aufbau des abstrakten Syntaxbaums)

  Codesharing: Git-Repository über github: https://education.github.com
- **Semantische Analyse:** Typisierung der abstrakten Syntax **(1 Person)**
- **Codeerzeugung (3 Personen):**
  - Aufbau eines abstrakten ClassFiles (1 Person)
  - Konstantenpool (1 Person)
  - (Umwandlung des ClassFiles in Bytecode, (1 Person))
- **Tester (3 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)
  - Händische Typisierung aller Testfälle der abstrakten Syntax in abstrakten Bytecode (als Test–Eingaben für den Code-Generierer)
  - Händische Übersetzung aller Testfälle des abstrakten Bytecodes in konkreten Bytecode (.class-Files)
  - Automatische Tests, die die jeweilgen Testsuite mit den implementierten Funktionen des Teams vergleichen

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 erbrachten 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**