

Praktikumsaufgabe 1 Compilerbau

Aufgabe 0 (Einführung)

Laden Sie die Praktikumsressourcen von Moodle herunter. Lesen Sie das ReadMe der heruntergeladenen Ressourcen. Es enthält wichtige Informationen zur Implementierung Ihres Compilers.

Aufgabe 1 (Sprachdefinition)

Machen Sie sich mit unserer Sprache SPL vertraut. Studieren Sie dazu zuerst die SPL-Sprachdefinition im Praktikumsskript.

Aufgabe 2 (Programmieren in SPL)

Vertiefen Sie ihre SPL-Kenntnisse, indem Sie mindestens 5 kleine SPL-Programme schreiben. Versuchen Sie, möglichst viele Features der Sprache zu benutzen. Hier ein Vorschlag:

- Im Hauptprogramm: Eingabe von zwei Zahlen, Ausgabe der Summe.
- Dasselbe, aber mit Berechnung der Summe in einer Prozedur.
- Einlesen eines Arrays von Zahlen im Hauptprogramm, Sortieren des Arrays in einer Prozedur, Ausgabe des sortierten Arrays im Hauptprogramm.
- Berechnung von $n!$ in einer Prozedur, sowohl iterativ als auch rekursiv.

Hinweis: Um Ihre Programme auf syntaktische Richtigkeit zu testen, können Sie sie den Tutoren vorlegen. Außerdem steht Ihnen auf Moodle in den Eco32-Tools eine Referenzimplementierung für Linux zur Verfügung.

Aufgabe 3 (Scanner-Implementierung mit Scanner-Generator)

Schreiben Sie die Spezifikation eines Scanners für SPL in Form einer Eingabedatei für den Scannergenerator.

Ergänzen Sie dazu die folgende Datei:

C: `src/phases/_01_scanner/scanner.flex`

Java: `src/main/java/de/thm/mni/compilerbau/phases/_01_scanner/Scanner.flex`

Aufgabe 4 (Testen des Scanners)

Lassen Sie Ihren Scanner die 5 Testprogramme aus dem ersten Aufgabenteil analysieren. Dazu starten Sie den Scanner mit der Option „- -tokens“.

Beobachten Sie den Tokenstrom und korrigieren Sie ggf. Ihre Scanner-Spezifikation.

Aufgabe 5 (Test mit allen in SPL definierten Tokenarten)

Schreiben Sie eine Eingabedatei zum systematischen Test ihres Scanners, vor allem in Hinblick auf die verschiedenen Formen von Ganzzahl-Literalen.