

Compilerbau - Praktikumsaufgabe 1 „SPL Scanner“

Aufgabe 1 (Sprachdefinition)

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

Aufgabe 2 (Programmierung 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:

- a) Im Hauptprogramm: Eingabe von zwei Zahlen, Ausgabe der Summe.
- b) Dasselbe, aber mit Berechnung der Summe in einer Prozedur.
- c) Einlesen eines Arrays von Zahlen im Hauptprogramm, Sortieren des Arrays in einer Prozedur, Ausgabe des sortierten Arrays im Hauptprogramm.
- d) 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 dem MNI-Rechner 'saturn' unter dem Pfad `/home/hg52/spl` eine Referenzimplementierung für Linux zur Verfügung. Die Referenzimplementierung können Sie entweder mit `ssh`-Login direkt nutzen oder mit `scp` oder `rsync` auf ihren eigenen Rechner kopieren und dort verwenden.

Aufgabe 3 (Scanner-Implementierung mit Scanner-Generator)

Schreiben Sie die Spezifikation eines Scanners für SPL in Form einer Eingabedatei für den Scanner-generator.

Ergänzen Sie dazu die Datei `scanner.jflex` (C-Lösung: `scanner.1`) und fügen Sie eine Möglichkeit zur textuellen Ausgabe von Tokens (ggf. mit Wert) hinzu.

Aufgabe 4 (Testen des Scanners)

Lassen Sie Ihren Scanner die 5 Testprogramme aus dem ersten Aufgabenteil analysieren. 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.