

Übungen zur Vorlesung: Struktur und Implementierung von Programmiersprachen II Blatt 1 (Abstrakte Syntaxbäume)

Weil Parsing bereits Inhalt von SIPS I war, ist der Parser vorgegeben. Dieses Blatt dient dazu, sich mit dem Aufbau der abstrakten Syntaxbäume (AST) für den MiniJava-Compiler sowie mit dem Visitor-Pattern vertraut zu machen. Mehrere (Teil-)Phasen des Compilers werden unter Verwendung dieses Patterns implementiert.

Hinweise:

1. Gruppenarbeit mit bis zu drei Personen ist zu empfehlen.
2. Die Webseite für Vorlesung und Übung lautet:
<http://infosun.fmi.uni-passau.de/cl/lehre/sips2-ws0607/index.html>
3. Schicken Sie bitte Ihr Programm in gepackter Form einen Tag vor der Übung an herrmann@uni-passau.de, mit dem Subject: SIPS2.

Aufgabe 1 (Verbesserung des Prettyprinters)

Auf der Webseite finden Sie einen Prettyprinter für MiniJava, implementiert in Java. Importieren Sie die Dateien in Eclipse und testen Sie zunächst, ob Parser und Prettyprinter die Beispielprogramme verarbeiten können.

- (a) Wie Sie an den Quelldateien erkennen können, werden Accept- und Visit-Methoden mit unterschiedlichen Typsignaturen benötigt. Definieren Sie ein generisches Interface `GenVisitor<A,R>` für Visit-Methoden mit Argument vom Typ A (zusätzlich zum `this`-Pointer) und Rückgabewert vom Typ R und ergänzen im Package `syntaxtree` die generischen Accept-Methoden.
- (b) Verbessern Sie die Ausgabe des Prettyprinters so, dass die Rümpfe von Schleifen und Zweige von Fallunterscheidungen je nach Schachtelungstiefe unterschiedlich weit eingerückt sind. Verwenden Sie nach Möglichkeit das Ergebnis von Teilaufgabe (a), indem Sie das Interface `GenVisitor<A,R>` mit der Instanzierung `A=Integer` (Einrücktiefe) und `R=Void` geeignet implementieren.

Besprechung: in der Übung am Donnerstag, 26.10.2006.