

	<h1>Universität Karlsruhe (TH)</h1> <h2>Lehrstuhl für Programmierparadigmen</h2>		
	Compilerpraktikum SS 2009 Dozent: Prof. Dr.-Ing. G. Snelting Betreuer: Matthias Braun Betreuer: Jürgen Graf	http://pp.info.uni-karlsruhe.de/ snelting@ipd.info.uni-karlsruhe.de braun@ipd.info.uni-karlsruhe.de graf@ipd.info.uni-karlsruhe.de	
	Übungsblatt 4	Ausgabe: 13.05.2009	Besprechung: 20.05.2009

Ziel in dieser Woche ist die Entwicklung der semantische Prüfung. Die semantische Analyse holt Dinge nach, die wegen der beschränkten Grammatikklassen nicht während des Parsens erledigt werden können. Dazu gehören Dinge wie Namensanalyse und Typprüfung.

Im allgemeinen gilt: Liefert diese Prüfung keine Fehler so muss der Compiler (bis auf Resourcebeschränkungen) in der Lage sein das Programm zu übersetzen.

Aufgabe 1: Nachtrag

Unter http://pp/lehre/SS2009/compprakt/wiki/index.php/Tests_Syntax befindet sich eine wiki-Seite auf der wir Beispielprogramm mit korrekter und inkorrekt Syntax sammeln. Bitte stellt dort ein paar Beispiele ein, falls ihr das noch nicht getan habt. (Um Dateien hochzuladen muss man sich einen wiki-Account anlegen).

Aufgabe 2: Namensanalyse

- Welche Elemente im AST müssen verlinkt werden? Welche Namensräume gibt es? Wie sind die Namensräume verschachtelt?
- Entwerft eine Schnittstelle zur Namensanalyse während eines AST-Durchlaufs und implementiert diese.
- Benutzt eure Namensanalyse um Deklarationen mit ihren Benutzern zu verbinden und legt die Ergebnisse im AST ab.

(Wir empfehlen sich die Vorlesungsfolien und Übungsblatt 6 der Vorlesung „Sprachtechnologie und Compiler“ aus dem Wintersemester 2008 anzusehen).

Aufgabe 3: Semantische Prüfung - Typen

- Welche Typen existieren in Minijava? Braucht man weitere Typen?
- Entwerft Typisierungsregeln für die Sprache: Welche Typen sind als Inputs von Operationen erlaubt und welche Typen haben dann die Ergebnisse?
- Implementiert diese Typisierungsregeln.

Aufgabe 4: Semantische Prüfung - Sonstiges

- Welche weiteren Tests sind in der Semantikphase nötig? Implementiert diese.

Aufgabe 5: Testen

- Schreibt Testprogramme die keine Syntaxfehler enthalten aber typische semantische Fehler provozieren.

Tauscht eure Testprogramme über das Wiki mit den anderen Gruppen aus:

http://pp/lehre/SS2009/compprakt/wiki/index.php/Tests_Semantik