

Kurze Beschreibungen

Aufgabe 1:

In Aufgabe 1 wurde für die gegebene Grammatik in EBNF-Form ein Parser geschrieben. Da diese Grammatik keine LL(1) Grammatik ist, sondern ein Lookahead von 2 erforderlich ist, wurde dieser erhöht.

Das jj-File für Aufgabe 1 befindet sich unter „src/yapl/impl/ca2_1/CA2_1.jj“

Aufgabe 2 (Scanner):

Für die Implementation des Scanners werden die Tokens benötigt, welche wir in „CA2_2.jj“ hinzugefügt haben. Der Scanner sollte von einem File lesen, was bei uns über die Konsole passiert. Ein Special Token für Kommentare im Quellcode wurde hinzugefügt.

Das jj-File befindet sich unter „src/yapl/impl/scanner/CA2_2.jj“.

Es gibt ein ant-target um den scanner mit allen yapl-files einer Compilerversion aufzurufen, nämlich „**ant run-scanner-all**“.

„run-scanner-all“ funktioniert ganz gleich wie „eval-all“, nur dass nicht „eval-compiler-msg“ aufgerufen wird, sondern „run-scanner“, welches den Scanner für das File im property „{yapl}“ aufruft. Das File, in welches der Scanner seinen output schreibt, ist im property {log} gespeichert.

„run-scanner“ ist wiederum abhängig von „compile-scanner“, welches die CA2_2.java – Klasse kompiliert und das class file im directory in {build} ablegt.

„compile-scanner“ ist abhängig von „mkdirs“, welches das directory in {build} anlegt.

Außerdem ist „compile-scanner“ abhängig von „compile-scanner-jj“.

„compile-scanner-jj“ führt mithilfe der von uns eingeführten Properties „scannerdir“, „scannerfile“ und „scannerclass“ das javacc command für CA2_2.jj aus.

Aufgabe 3(Parser):

Für den Parser werden nun auch die Produktionen benötigt. Diese wurden in „CA2_3.jj“ hinzugefügt. Der Parser gibt eine Output-Message mit dem Programmnamen und einem OK aus, wenn das Inputfile ein korrektes YAPL-Programm darstellt. Ist das Inputfile kein korrektes YAPL-Programm, so werden entsprechende Fehler ausgegeben.

Das target zur Ausführung des Parsers mit allen Testfiles einer Compilerversion ist „eval-all“, welches aber eh als default target definiert ist. Deshalb reicht „ant“.

Zusätzliche Properties für den Parser sind „parserdir“ (Directory, welches die jj-Datei enthält), „parserfile“ (Pfad zur jj-datei) und parserclass (Klassenname inklusive Packages, also hier „yapl.compiler.CA2_3“).

Zusätzlich eingeführtes ant-target ist „mkdirs“ welches das build-directory erstellt.

Das jj-file für den compiler befindet sich unter src/yapl/compiler/CA2_3.jj

Sonstiges:

Ausführung mittels **ant**.

Der default-Wert in built.xml wurde auf eval-all gesetzt.

Es müssen eigentlich keine ant-properties überschrieben werden, da alle Dependencies im resources-folder mitgeliefert werden. Ausführen von „ant“ reicht für den Parser. Ausführen von „ant run-scanner-all“ für den Scanner mit allen testfiles.