

Compiler-Compiler-Komponenten II

(Konsolidierungsphase)

In dieser Phase sollen die Ergebnisse der beiden vorangegangenen Phasen

- *abgerundet*
- überarbeitet und
- dokumentiert

werden.

Vordringlicher Zweck dieser Phase ist es, alles bestehende Material in eine wohl-dokumentierte, ordentliche Form zu bringen, so daß eine Basis entsteht, auf der man weiterarbeiten kann.

Die Ergebnisse sollen schriftlich in HTML (Ruby-Programme mit Testdateien etc in zip-Files) geliefert (von mir in Homepage integriert) werden. Die mündliche Präsentation soll durch die Autoren erfolgen, wobei Fragen zum Vortrag, zur Ausarbeitung und zum theoretischen Hintergrund die mündliche Prüfung ergeben.

Die **Gruppeneinteilung** ist wie in der vorangegangenen Phase, mit einer Ausnahme: die Scannergruppe hat wieder die Zusammensetzung aus der Einarbeitungsphase.

Die Themenblöcke sollen wie folgt **abgerundet** werden:

- Das **Editorfenster** erhält eine neue Funktion: Erstellung eines „Recursive Descent Parsers“. Anhand der LL(1)-Entscheidungsmengen läßt sich der Text eines Recursive Descent Parsers einfach erzeugen (siehe S 177f.)
- Als letzter **Parser**-Typ wird der LALR(1)-Automat (durch Verschmelzen der kerngleichen LR(1)-Zustände, siehe S. 166) berechnet.
- Analog zur Jaccie-**Scanner**-Beschreibung sollen Jaccie-Grammatik-Beschreibungen gelesen werden, wobei eine Grammatik als Objekt der gleichnamigen Ruby-Klasse entsteht.

Alle Arbeitsergebnisse (dieser und insbesondere der vorangegangenen Phasen) sollen sorgfältig geglättet und dokumentiert werden, Diese Arbeit soll von den jeweiligen Autorengruppen geleistet werden

Mit „**Glätten**“ meine ich:

- Programmstruktur sinnvoll umbauen (Module bilden, importieren, Unit-Tests in eigene Testklassen auslagern, ...)
- Implementierung überarbeiten: Alles Überflüssige entfernen, ebenso umständliche Konstruktionen durch bessere ersetzen, Stil verbessern, alles selbstkritisch überdenken und verbessern.

Mit „**Dokumentieren**“ meine ich (alles in HTML):

- RDoc-Doku vervollständigen und überarbeiten. Dazu ein inhaltlicher Überblick über die Programmstruktur und die Doku-Teile.
- Überblick zu gewählten Unit-Tests
- Je Teilprogramm ein Kurzhandbuch mit Bedienungsanleitung. Dieser Teil ist beim Editorfenster ausführlicher zu halten.
- Zu allen Teilprogrammen sinnvolle Anwendungsbeispiele bearbeiten und dokumentieren. Für die Scannergruppe ergeben die von Hand implementierten Scanner aus der Einarbeitungsphase und der miniDoc-Scanner solche Anwendungsbeispiele ab. Von der Parser-Gruppe sollen Grammatiken verwendet werden, die die Grammatikklassen voneinander unterscheiden, und dazu die miniDoc-Grammatik als praktisches Beispiel.

Für **mündlichen Präsentationen** stehen die Termine bis zum Ende des Trimesters zur Verfügung.

- Es ist sinnvoll, die Präsentationen *aufgabenbezogen* durchzuführen, wobei einzelne mehrfach, mindestens aber einmal präsentieren müssen.
- *Anmeldungen* zu Präsentationen *frühzeitig*, ab sofort möglich!
- *Schriftliche Ergebnisse* müssen mir spätestens zum Mittwochabend vor dem jeweiligen Präsentationstermin vorliegen.