

Fortgeschrittenen-Praktikum

Sommersemester 2002

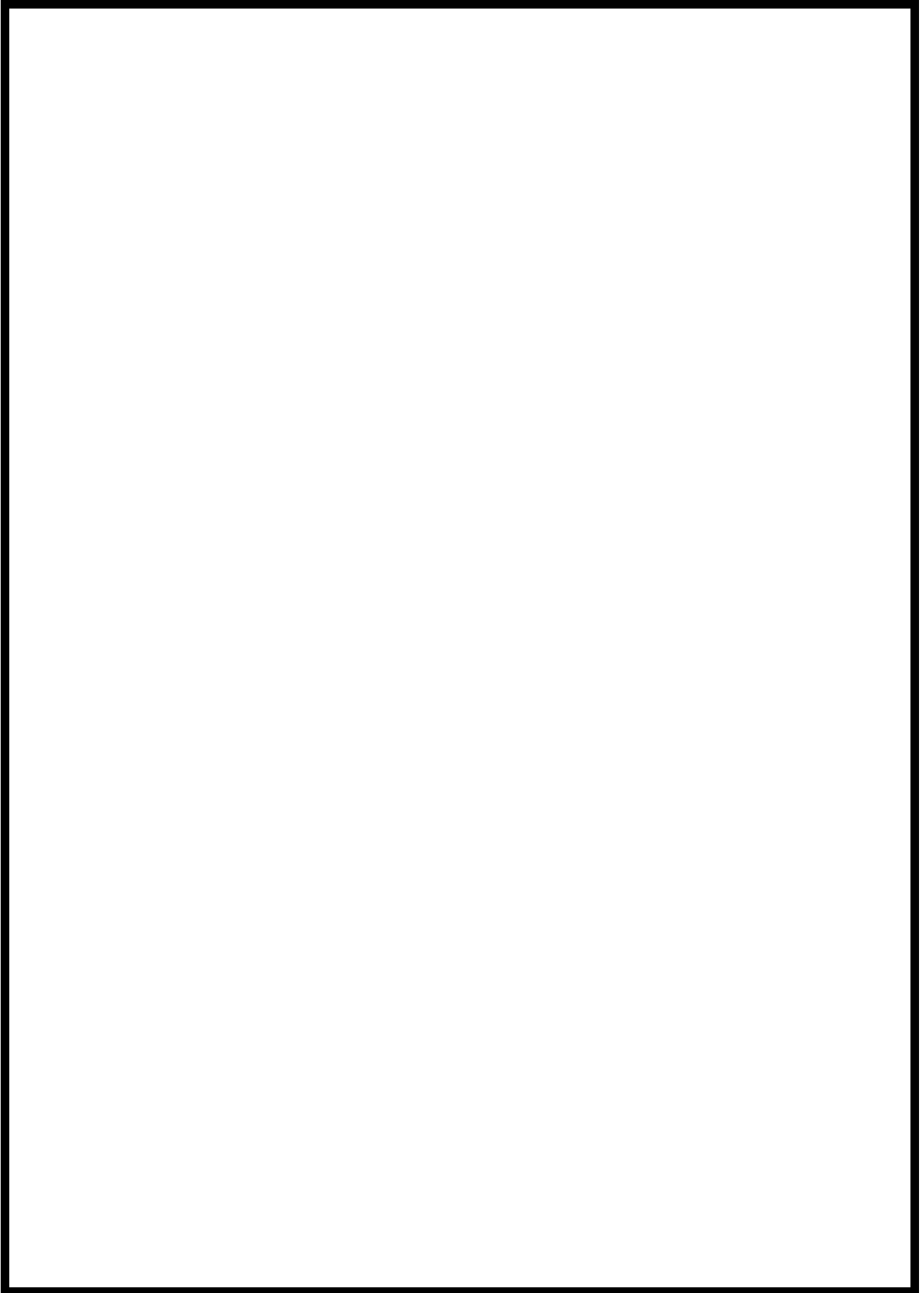
Programming-In-The-Many

Marco Wendel, #332972

17. 07. 2002

Modul SFCParser (Teilmodul von Slime)

- JLex als Scannergenerator
- CUP als Parsergenerator
- zwei unterschiedliche abstrakte Syntaxbäume
- mehrstufiges Vorgehen
- Behandlung geschachtelter Anweisungen



JLex

- JLex: A Lexical Analyzer Generator for Java(TM)
<http://www.cs.princeton.edu/~appel/modern/java/>
- Unterstützung von CUP
- Eingabedatei: input.lex
- Ausgabedatei: input.lex.java
 - enthält standardmäßig eine Klasse Yylex
 - Yylex liefert die sogenannten *Token*

CUP

- CUP Parser Generator for Java(TM)

<http://www.cs.princeton.edu/~appel/modern/java/CUP/>

- CUP = Constructor of Useful Parser
- Eingabedatei: parser.cup
 - Javacode des Benutzers
 - Symbolliste (terminale/nichtterminale Token)
 - Grammatik in BNF-naher Notation
- Ausgabedatei: parser.java, sym.java
 - werden durch java_cup.main erzeugt
 - benötigen java_cup.runtime.*

Abstrakte Syntaxbäume

- 2-stufiges Vorgehen um SFC-Baum zu erhalten
- Parsen in `slime.absynt.absfc.SFCabtree`
 - orientiert sich an konkreter Syntax
- Konvertierung von `SFCabtree` in `slime.absynt.SFC`
- 1. Prozesse, 2. Nicht-Prozesse und Zusammensetzung
 - rekursives Vorgehen über Statementlisten
 - *processStatementList* findet richtige Methode

Bewertung des Parsers

- bedingt durch JLex/CUP kaum Fehler bei der Tokenerzeugung aus Eingabedateien
- leicht zu modifizieren, da 2-stufig
- Aufwand: $O(n)$
- Fehlerbehandlung unbefriedigend, da die CUP-Fehlerbehandlung keine Exceptions vorsieht
- Leider kein Feld-Datentyp vorgesehen.