CS 565 November 14, 1996
Jens Palsberg

Term Project : Java Analyzer

The goal of the project is to write a Java Analyzer. The Java Analyzer annotates Java programs. The annotations are useful for optimizing compilers, programming environments, etc.

The input language is not full Java, but a subset called babyJava. On the CS 565 homepage you will find a lex file, a yacc file, and a Makefile for babyJava. A babyJava program is one which both can be accepted by the babyJava parser and can be compiled by the Java compiler.

babyJava = Java
- packages
- multithreading
- exception handling
- casting
- arrays
- some of the boring expression syntax

The Java Analyzer removes all comments from the input program and inserts an annotation, in the form of a comment, for each message send, declaration of a variable, and declaration of a method argument. Each annotation is a "concrete type", that is, a type in the sense of *OOTS* Chapters 4–5. The output of the Java Analyzer is a pretty-printed and annotated babyJava program.

The Java Analyzer takes its input from standard input and produces its output on standard output.

To do the static analysis of babyJava programs, use the `solver` data structure from Homework 9.

The project is due Wednesday November 27, 12 Noon. Already Friday Nov 22, 12 Noon all groups must submit to Hae-Young Lee a babyJava program that begins with a short comment which explains why this program is a good test of some aspect of the Java Analyzer. Shortly thereafter, we

will create a directory with all these programs such that everybody can use them for testing. Submitting the testprogram counts towards the grade for the project.

On Wednesday November 27, submit:

- Your program
- A report which describes your implementation. The report should assume that the reader is an expert on the contents of *OOTS*. You must therefore describe in detail what you do for constructs that are not (essentially) covered by *OOTS*. The report should also describe the overall structure of your implementation.

The grade will be determined on the basis of (1) your testprogram, (2) your report, (3) the correctness of the Java Analyzer, and (4) the beauty of the pretty-printed babyJava programs. The beauty will be graded on the scale: Nice, Wierd, and Obfuscated. Nice is best.