CS 565 Out: April 6, 1998
Jens Palsberg Due: April 20, 1998

Term Project : Java Optimizer

The goal of the project is to write a Java Optimizer. The Java Optimizer must be written in Java. The input language is BabyJava, see the CS 565 webpage in the code/babyjava directory. The output language is Java. You can assume that each input can be compiled by javac. Your Java Optimizer must terminate on all legal inputs. The output is like the input, except that some message sends have been inlined.

The Java Optimizer inlines as many message sends as possible/reasonable. Some examples of this are given on the CS 565 webpage in the code/inlining directory. The idea is to run type inference (flow analysis) in the sense of OOTS Chapter 5, and then use the computed information as the basis for replacing message sends with possibly conditional statements where each branch is inlined code. There are many possible strategies for doing this, and your task is to invent one and implement it.

Ideally, the output language would be machine code. To save you from the trouble of writing all of an optimizing compiler, the output language is Java where you can take advantage of the "instanceof" expression to test the class of an object. The only optimization your Java Optimizer should do is inlining; you will not get credit (but certainly some goodwill :-) for implementing any of all the other possible optimizations like CPS-conversion, constant propagation, common subexpression elimination, etc.

The Java Optimizer is called as follows:

```
java Optimize < Prg.java > OptimizedPrg.java
```

where `Prg.java` is a file containing a BabyJava program.

Submit your program together with a README file which describes in detail your strategy for inlining. The description should both explain the idea you are using and also give at least one example.

Grading will be based mostly on what works, partly on how smart and elegant your inlining strategy is, and also, to some extent, on the quality of the writing of your README file.