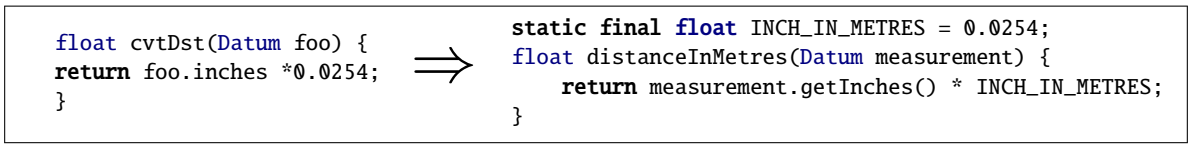
How does Refactoring affect Java performance?

Christoph Reichenbach, christoph.reichenbach@cs.lth.se

20. Dezember 2017

Wanted: One or two Java programmers with interest in refactoring and benchmarking. At least one of you should have taken the Compilers course. This is an M.Sc. thesis project supervised by Christoph Reichenbach in the SDE group.



Refactoring is a common strategy for making code more readable and more maintainable. However, refactoring may speed up or slow down the program, and some developers can be hesitant to employ refactoring for that reason. In this project, we will systematically analyse the impact of automated refactorings on Java program performance, both in end-user programs and in libraries, by performing refactorings (such as 'Extract Method', 'Inline Method', or 'Inline Temporary') on randomly selected program locations, or on 'bad smells' detected by Eclipse plugins such as 'The Spartanizer'. You can even implement and try out your own refactorings, if you are interested.

A central part (and the bulk of the work) for this project will be in assembling a benchmark suite of representative Java programs and libraries that can be refactored, compiled, tested, and benchmarked easily. There are existing suites that we can draw inspiration from, but none of them are completely suitable for our needs. For instance, none provide a good cover of Java 8 language features.

This thesis project entails:

- Learning about benchmarking and the existing Java benchmark suites.
- Collecting benchmarks (both applications and libraries) and making them easily *compilable*, *refactorable*, *testable*, and *runnable* on different platforms and with multiple Java implementations.
- Building a benchmark harness that allows us to easily try out different refactorings and gather performance statistics.
- Extending the benchmark suite to ensure that it covers all Java 8 language features, using the SDE group's ExtendJ system.
- Picking a framework for refactoring (such as Eclipse or NetBeans) and writing code to automatically apply the existing refactorings provided by that framework or by existing plugins to that framework.
- Evaluating the performance impact of these refactorings.
- *Optionally:* building your own refactorings and exploring their performance impact.
- Writing the thesis.

If you are interested, feel free to contact me at christoph.reichenbach@cs.lth.se.