



ALBERT-LUDWIGS-  
UNIVERSITÄT FREIBURG

J. Hoenicke  
J. Christ

08.02.2012

Hand in solutions via email to  
`christj@informatik.uni-freiburg.de`  
until 15.02.2012 (only Java sources, KeY  
proofs, and PDFs accepted).  
Paper submissions possible after the lecture.

## Tutorials for “Formal methods for Java” Exercise sheet 13

This exercise sheet deals with the MyStack example presented in the lecture. The source file for this example can be downloaded from the website. To run it with JPF, place it in the examples directory of the JPF project you created on the previous exercise sheet, compile it with ant, write a configuration for it, and run JPF.

### Exercise 1: Detecting Shared Objects

Partial order reduction as implemented in JPF relies on a detection of objects that are shared between multiple threads. An algorithm for this detection was presented in the lecture. Apply this algorithm to the program after the second thread has been started. The root set at this time contains the main thread, and a thread for each MyStack. For each shared object, state the threads that can access this object.

### Exercise 2: Fixing the Code

Fix the code, i.e., remove all bugs in the code that JPF can find.

### Exercise 3: Branch Coverage

JPF provides a coverage analyzer. This tool can be used to calculate different coverage metrics. Use this tool to compute the branch coverage for the corrected version. You may need the configuration option `coverage.include`.

### Exercise 4: Improving Branch Coverage

The previous exercise did not show 100% branch coverage. Rerun the coverage analysis with the options `coverage.show_methods`, and `coverage.show_bodies` enabled. Deduce from the output a new test case that achieves 100% branch coverage. Fix the code to get 100% branch coverage.