

Formal Methods in Software Engineering

Exercise 6 (December 15)

Wolfgang Schreiner
Wolfgang.Schreiner@risc.uni-linz.ac.at

November 18, 2008

The result is to be submitted by the deadline stated above via the Moodle interface as a .zip or .tgz file which contains

- A PDF file with
 - a cover page with the title of the course, your name, Matrikelnummer, and email-address,
 - for each exercise, a section with the number and name of the exercise, the JML-annotated Java code, and a copy of the output of an `escjava2` check of that code,
 - optionally any explanations or comments you would like to make;
- the JML-annotated Java files developed in the exercise.

6a (50 points): JML function specifications

Annotate the functions in class `Exercise6a` with JML header specifications that are as expressive as possible.

Type-check the specifications with `jml -Q` (which must not give an error) and statically check them with `escjava2` (which may or may not give warnings).

Please take care to constrain the arguments by appropriate preconditions. The fact that above functions do not modify the content of argument *a* can be specified by the JML clause `assignable \nothing`.

6b (40 points): JML procedure specifications

Annotate the functions in class `Exercise6b` with JML header specifications that are as expressive as possible.

Type-check the specifications with `jml -Q` (which must not give an error) and statically check them with `escjava2` (which may or may not give warnings).

Please take care to constrain the arguments by appropriate preconditions. The fact that above functions modify the content of argument a can be specified by the JML clause `assignable a[*]`.

6c (10 points): JML exception specification

Annotate the function in class `Exercise6c` with a *heavy-weight* JML header specification that is as expressive as possible. Type-check the specification with `jml -Q` (which must not give an error) and statically check it with `escjava2` (which may or may not give warnings).

Bonus (10 points)

(At least) one of the Java functions has an error. Demonstrate with the help of `escjava2` the error, fix it, and show the output of `escjava2` on the new code.