

# Formal Methods in Software Development

## Exercise 6 (January 7)

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

November 24, 2007

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, and, for each exercise,
  - the JML-annotated Java code,
  - a screenshot of KeY after completion of the proof,
  - an explicit statement whether the proof was completed successfully or whether there were open proof situations left,
  - for each open proof situation, a KeY screenshot and an (attempt of an) explanation why the proof did not go through,
  - optionally any explanations or comments you would like to make.
- the JML-annotated Java code (.java file)
- the KeY proof of the verification (saved as an .proof file).

### 6a (KV4): Searching for the Minimum

Take the JML-specified Java method

```
// returns minimum of non-empty array a
static int minimum(int[] a)
```

from Exercise 3 that returns the minimum element of a non-empty array  $a$ .

Annotate the loop in its body with a suitable invariant (`loop_invariant`), termination term (`decreases`), and frame condition (`assignable`) and verify with KeY the method's total correctness (proof obligation for normal behavior only).

## 6b (KV4): Replacing Array Elements

Proceed as in the previous exercise with the method

```
// replaces in a every occurrence of x by y
static void replace(int[] a, int x, int y)
```

from Exercise 3 that replaces in  $a$  every occurrence of  $x$  by  $y$ .

---

As shown in class, please explicitly add (in both parts of the exercise) a condition on `a.length` to the method precondition respectively loop invariant.

If some proof should not be completely successful, minimize the number of open goals as far as possible and add for each open proof situation a KeY screenshot and an (attempt of an) explanation.