J. Hoenicke
J. Christ

27.11.2012
Hand in solutions via email to
`christj@informatik.uni-freiburg.de`
until 04.12.2012 (only Java sources, JPF configuration files, and PDFs accepted).
Paper submissions possible after the lecture.

Tutorials for "Formal methods for Java"
Exercise sheet 6

Exercise 1: Installing and Configuring jpf-aprop

Get a copy of the jpf-aprop repository either with

```
hg clone http://babelfish.arc.nasa.gov/hg/jpf/jpf-aprop
```

or a similar command from your Mercurial client.
Compile the downloaded version, e.g., using ant from the cloned repository:

```
bin/ant
```

Add a line

```
jpf-aprop=<path/to/jpf-aprop>
```

to your `.jpf/site.properties`. Replace the part between < and > with the correct path.
Create a new JPF project that depends on jpf-aprop using jpf-template with the following command line

```
<path-to-jpf-template>/bin/create_project <Project-Name> jpf-aprop
```

Again, replace the part between < and > with the correct path.
You don't have to submit anything for this exercise.

Exercise 2: InsertionSort
Consider the following code for an insertion sort algorithm.

```
class InsertionSort {
   public static void sort(int[] arr) {
      for(int i = 1; i < arr.length; i++) {
         for(int j = i; j > 0; j--) {
            if (arr[j] <= arr[j-1]) {
               int tmp = arr[j];
               arr[j] = arr[j-1];
               arr[j-1] = tmp;
            }
         }
      }
   }
}
```

Write a minimal specification in the Syntax of jpf-aprop. Your specification should prevent the implementation from crashing and ensure the array is sorted in ascending order of the keys after the function terminates.
Write a test and a JPF configuration file to test your annotations with jpf-aprop. You should put these files in the `src/examples` folder of your newly created project. Hand in the test, the configuration, and the annotated Java-Program.

Exercise 3: ESC/Java 2 installation
Download and install ESC/Java 2 from
`http://kindsoftware.com/products/opensource/ESCJava2/download.html`.
You don't have to hand in anything for this exercise.

Exercise 4: Insertion Sort (cont.)
Consider again the insertion sort algorithm from Exercise 1. Reformulate the specification from that exercise in JML and verify it with ESC/Java 2. Hand in your annotated program.