J. Hoenicke
J. Christ

08.01.2013
Hand in solutions via email to
`christj@informatik.uni-freiburg.de`
until 15.01.2013 (only Java sources and
PDFs accepted).
Paper submissions possible after the lecture.

Tutorials for “Formal methods for Java”
Exercise sheet 9

Exercise 1: Jahob Syntax

Translate the JML annotations in the following program into Jahob syntax.

```
class IntKey {
    int value;
}

public class BubbleSort {
    /*@ requires arr.length > 0 && (\forall int i; i >= 0 && i < arr.length; arr[i] != null);
      @ ensures (\forall int k, l; 0 <= k && k <= l && l < arr.length;
      @                            arr[k].value <= arr[l].value);
      @*/
    public void sort(/*@ non_null */ IntKey[] arr) {
        /*@ loop_invariant i >= 0 && i < arr.length;
          @ loop_invariant (\forall int i; i >= 0 && i < arr.length; arr[i] != null);
          @ loop_invariant (\forall int k, l; i <= k && k <= l && l < arr.length;
                                    arr[k].value <= arr[l].value);
          @ loop_invariant (\forall int k, l;
          @                 0 <= k && k <= i && i < l && l<arr.length;
          @                 arr[k].value <= arr[l].value);
          @
          @*/
        for (int i = arr.length−1; i > 0; i−−) {
            /*@ loop_invariant i >= 0 && i < arr.length;
              @ loop_invariant (\forall int i; i >= 0 && i < arr.length; arr[i] != null);
              @ loop_invariant j >= 0 && j <= i;
              @ loop_invariant (\forall int k, l; i <= k && k <= l && l < arr.length;
              @                                   arr[k].value <= arr[l].value);
              @ loop_invariant (\forall int k, l;
              @                 0 <= k && k <= i && i < l && l<arr.length;
              @                 arr[k].value <= arr[l].value);
              @ loop_invariant (\forall int k; 0 <= k && k < j;
              @                                arr[k].value <= arr[j].value);
              @*/
            for (int j = 0; j < i; j++) {
                if (arr[j].value >= arr[j+1].value) {
                    IntKey tmp = arr[j];
                    arr[j] = arr[j+1];
                    arr[j+1] = tmp;
                }
            }
        }
    }
}
```