

```
1 public class TickTockClock {
2   //@ public model JMLDataGroup _time_state;
3
4   //@ protected invariant 0 <= hour && hour <= 23;
5   protected int hour; //@ in _time_state;
6   //@ protected invariant 0 <= minute && minute <= 59;
7   protected int minute; //@ in _time_state;
8   //@ protected invariant 0 <= second && second <= 59;
9   protected int second; //@ in _time_state;
10
11  //@ ensures getHour() == 12 && getMinute() == 0 && getSecond() == 0;
12  public /*@ pure */ TickTockClock() {
13    hour = 12; minute = 0; second = 0;
14  }
15
16  //@ requires true;
17  //@ ensures 0 <= \result && \result <= 23;
18  public /*@ pure */ int getHour() { return hour; }
19
20  //@ ensures 0 <= \result && \result <= 59;
21  public /*@ pure */ int getMinute() { return minute; }
22
23  //@ ensures 0 <= \result;
24  //@ ensures \result <= 59;
25  public /*@ pure */ int getSecond() { return second; }
26
27  /*@ requires getSecond() < 59;
28    @ assignable hour, minute, second; // NB for expository purposes only
29    @ assignable _time_state;
30    @ ensures getSecond() == \old(getSecond() + 1) &&
31    @ getMinute() == \old(getMinute()) &&
32    @ getHour() == \old(getHour());
33    @ also
34    @ requires getSecond() == 59;
35    @ assignable _time_state;
36    @ ensures getSecond() == 0;
37    @ ensures (* hours and minutes are updated appropriately *);
38  */
39  public void tick() {
40    second++;
41    if (second == 60) { second = 0; minute++; }
42    if (minute == 60) { minute = 0; hour++; }
43    if (hour == 24) { hour = 0; }
44  }
45 }
```

Fig. 1. JML specification for TickTockClock. The datagroup `_time_state`, the associated `assignable` clauses and `in` clauses are explained later, in Section 3.