The University of Waikato
Department of Computer Science
Dr Tim Stokes and Dr Robi Malik

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

COMP 340-08B

Reasoning about Programs

Notes on Using the JML Tools

Before you Start

The use of the JML tools requires you to downgrade your Java version to 1.4. Open a shell window and enter the command `java-setup`. Then, after having entered your password, select option

```
2  /opt/j2sdk1.4.2_13/bin/java
```

by typing its number `2` and pressing ⟨return⟩.

Assertion Syntax

Assertions are introduced into Java programs using special comments marked by the `@` sign:

`//@` ... *single-line assertion* ...
`/*@` ... *multi-line assertion block* ... `*/`

Most assertions start with a keyword, are followed by a formula, and terminated by a semicolon. The most important assertions are as follows.

`requires` ⟨*formula*⟩`;` Defines ⟨*formula*⟩ to be a precondition of the method following the assertion block.

`ensures` ⟨*formula*⟩`;` Defines ⟨*formula*⟩ to be a postcondition of the method following the assertion block. In the formula, the special symbol `\result` can be used to refer to the return value of the method.

`assert` ⟨*formula*⟩`;` Defines a formula that is always supposed to be true at the point in the program where it appears. The JML Runtime Assertion Checker will abort with an error if the formula is ever found not to hold, and the Extended Static Checker will try to verify it.

`assume` ⟨*formula*⟩`;` Like `assert`, this defines a program assertion that is always supposed to be true. Only it is treated differently by the Extended Static Checker, which will not try to prove this formula, but assume it to be true in its attempts to prove other formulas.

JML Runtime Assertion Checker

The JML Runtime Assertion Checker executes a Java program with JML annotations, checking all the annotations while the program is running. Obviously, if your program contains lots of assertions (as it should), this will slow down execution considerably, but you can always switch back to a normal Java compiler once you are satisfied that the program is correct.

To use the JML Runtime Assertion Checker, you must use the JML Compiler `jmlc` instead of `javac` to compile your program, and run it using the JML Runtime System `jmlrac` instead of `java`. To compile and check the example program, you can use the following commands.

```
/home/robi/bin/jmlc -Q ArrayTests.java
/home/robi/bin/jmlrac ArrayTests 100
```

Note. Without the `-Q` option, the `jmlc` compiler produces a huge amount of output, which is helpful to confirm progress; but the output obliterates error messages.

Extended Static Checker for Java

The Extended Static Checker `escj` is a tool that reads a Java program with JML annotations and tries to verify them using the rules of Hoare logic. Unfortunately, the verification engine is as yet very incomplete, so the tool will fail to find all but the simplest proofs. Nevertheless, it is a very useful tool that can point out many indexing or null pointer errors without actually running a program.

To use the Extended Static Checker to analyse the example program, use the following command.

```
/home/robi/bin/escj ArrayTests.java
```

Due to its incompleteness, the Extended Static Checker may fail to complain about incorrect assertions, and it may report that it is unable to prove an assertion that is actually correct. In the latter case, if you are definitely convinced that the assertion in question is correct, you may replace the keyword `assert` by `assume`. Then the Extended Static Checker will stop complaining about that assertion, and instead use it in its attempts to prove other assertions. The JML Runtime Assertion Checker will continue to check this assumption as before.

Note. This should not be necessary for your present assignment.

Documentation

For more documentation of the JML notation, or for download of the tools, please refer to the homepages of the JML project.

- The Java Modeling Language (JML) at `http://www.jmlspecs.org/`
- ESC/Java2 at `http://secure.ucd.ie/products/opensource/ESCJava2/`