

# Checking the `DataOutputStream` Java Standard Class

## Homework On JML

Néstor Cataño C.  
ncatano@puj.edu.co

You will be specifying the `java.io.DataOutputStream` standard Java class in JML, and checking your specifications using both The JML Tool and The ESC/JAVA 2 Tool. To carry out this specification and checking work you need to go through the following steps :

1. Install the Java Development Kit (JDK) tool 1.4.1 or 1.4.2 as well as the tool source code on your work station.
2. Install the JML and ESC/JAVA 2 tools on your favourite Operating System. The JML tool can be downloaded from <http://www.cs.iastate.edu/~leavens/JML/download.shtml>. The tool runs on both Linux and Windows. The ESC/JAVA 2 tool can be downloaded from <http://secure.ucd.ie/products/opensource/ESCJava2/download.html>. This tool runs on Windows, Linux and Mac. Check for installation instructions for each tool. If you use the Eclipse programming environment, notice that Eclipse plugins for ESC/JAVA 2 and JML tools exist already. Instructions about installing the ESC/JAVA 2 plugin for Eclipse can be found at <http://jmleclipse.projects.cis.ksu.edu/docs/esc-java-install.shtml> and for JML at <http://jmleclipse.projects.cis.ksu.edu/download.shtml>. The Eclipse tool can be reached at <http://www.eclipse.org/downloads/>.
3. Read on inheritance, abstract classes, overriding, and *checked* and *unchecked* exceptions in Java.
4. Read Sections 1 and 2 of the *Preliminary Design of JML* document found at <http://www.cs.iastate.edu/~leavens/JML/prelimdesign/prelimdesign.toc.html> for a clear complete presentation on the Java Modeling Language. Check the course web-page for slides and examples on specifying Java libraries in JML.
5. Specify the standard Java class `java.io.DataOutputStream` in JML. This class implements the `DataOutPut` interface and extends the `FilterOutputStream` class, which in turn extends the `OutputStream` class, so provide specifications for them as well.
6. When providing specifications, write `normal_behavior` specifications for every method, and `exceptional_behavior` specifications for methods throwing checked exceptions. Notice that you do not need to specify the exceptional behaviour for methods throwing unchecked exceptions.
7. Specify class invariants for `DataOutputStream`, `FilterOutputStream`, `DataOutput` and `OutputStream`.
8. Specify the `DataOutPut` interface before giving specifications for any other piece of code. Decide on how you want to model the information given by this interface. Use `model`, `represents` and `in` specifications to this purpose. Base your-own work on previous work on specifying Java code in JML. Check for instance the JML specifications for `DataInput` and `BoundedStack` included in the JML distribution.
9. Check your specifications using both JML and ESC/JAVA 2. Notice that the specification language used by ESC/JAVA 2 is not exactly JML but a subset of it.
10. Send me a copy of your specifications and a printout of the messages given by both The JML Tool and The ESC/JAVA 2 Tool. Your work will be judged on how complete and correct your specifications are.