Project 2 — ESC/Java2 Assignment

1. I think the code is now correct and it should work as expected. For instance, the method *add* (defined into the Amount class) now computes the right values when two Amounts are added and the removeAll method of the Bag class is also working even though it could have been implemented differently. Both programs should run without exceptions but, however, other issues may arise at run-time which the used tool didn't detect timely. As an example, consider what could happen if the sum of two Amount(S) exceeds the *Integer.*MAX_VALUE value. The result will be a negative integer and this will probably lead to other unintended behaviours.

2. One possible improvement of the specification language is to make easier the annotation process for the programmer/analyst. With large programs, code annotation may result in a big effort because one would have to (e.g.) annotate that a specific variable must be *non-null* for all the subroutines that handle it. One simple idea is annotate a variable only one time (e.g. when it's declared or after the instantiation) and provide the tool with the ability to identify where and how the value will be used at run-time.

3. One alternative tool is the free OpenJML tool, which can be downloaded at openjml.org. OpenJML doesn't perform annotation checking at compile-time only, it can provide run-time checking too! Using this tool will probably allow the analyst to detect more problems because of the more advanced technique it uses. Naturally, run-time annotation checking is not free. The mentioned tool has to do more work but has the advantage to find more issues and, at the same time, it may be as fast as ESC/Java2 to find the restricted set of problems that the latter is able to detect. Finally, the proposed tool should (in general) provide more confidence that the code is correct even though this will depend on the instance of the analysed program and on the annotations made.