Design document

Implementation

My implementation is done in Java, this is because it is the OO-language I am most familiar with. The structure revolves around the superclasses “Expression” and “Statement” which in turn have the functions “evaluate()” and “interpret()”. For simplicity I implemented “interpret()” to return true/false so that the “Stop” command in Robol can be used as “break” in Java, this way it can break out of loops as well as stop the robot. “evaluate()” returns int-values based on what type of Expression it is, and what is in the VarDecl-List that is sent in as a parameter. I hva chosen to simplify the “ArithExp” and “BoolExp” classes a bit by using a switch and a three parameter signature rather than making subclasses for each of the different types (Plus, Minus, Multiply, etc...). The code should be pretty easy to read and understand even though the only comments found is the BNF-grammar on top of each class.

In the code I have included the four test-programs that where given as examples in the oblig-text.