Sami Blevens

4/19/22

Project Progress Update

Based on my project plan that I submitted, I am mostly done with the project. I have completed updating TokenType.java to include a const token and updating the nextToken function along with Lexer testing. I have also completed, updating the parser and ast parser, as well as VarDeclStmt to include a isConst Boolean value. I also created a tuple class to use, and have completely updated the symbol table and type info files to use Map<String,Tuple<String,Boolean>> instead of Map<String,String>. From there I of course had to update all of the places in the static checker that utilize the .get or .add functions of both of those. Then I started updating the static checker to check if the value was const or not. I believe this is only needed in the assignStmt as you can't assign anything to a const variable, but you can use it in expressions and other places.

The only thing I have left to do is figure out how to determine if the function parameters are const or not. I was first thinking of doing this by adding the const variable when adding the args in the call functions. However the issue with this is that the static checker doesn't add the args to the symbol table or type info. It just checks that they are both the same types. So when the static checker goes through a function and checks the parameters, it will not know what value was passed in until the code generator, just what type the value must be. Because of this I am going to have an optional const that can be added in front of the functions arguments and if it is const, it will mark it in the symbol table / type info, and then the static checker will check first if the argument in the call expr is a const, then if the function definition marked that parameter as being constant. This is will not error if a non const value is passed into a function where the parameter is marked as constant, as it does not matter that a non constant value is not being changed. If a constant argument is being passed into a parameter where it is not mark as constant, this will error.

To complete that, I have to update the way the function arguments are statically checked, and possibly update the class variables to mark that. Otherwise, just input in the symbol table and type info that the parameter can be constant. Then update the callExpr where its already checking that the parameter types are the same.