CS371
Hw3: Typechecking II
Aashish Karki '15

Phase2.java goes through the input file for the second time and processes method bodies.

SymbolTable.java

To keep a record of variables and scopes, I am using a symbol table. The symbol table contains a hash map of local variables called localVarMap. currentList is a list that maintains list of variables in current scope. As the scope increases, currentList is pushed into a stack called stackList. As the scope of symbol table decreases, the stackList is popped and the popped list's variables are removed from the localVarMap. When a new variable is declared, it is added to currentList with current scope and also added to the localVarMap.

Addition to Typechecker.java:

I added some additional methods to Typechecker.java to handle method calls in method bodies. checkMethodHeaderAndArgs() checks the method being called exists and whether appropriate arguments are passed.

Phase2.java:

Added if conditions to various parts of scanner to detect Typechecking errors and throw Typechecker exceptions. I have extensively commented the code to help explain what errors I am trying to catch.

Phase2.java calls incScope() and decScope in SymbolTable at the start and end of various code blocks(e.g. AMethodMainDecl, AIfStmt, AWhileStme etc) respectively.

Phase3.java calls addEntry in SymbolTable when a new variable is declared.

I have added the Expr and ExprType classes as instructed and used them while processing method bodies.