Mohammadamin Sheikhtaheri - 0930853 March 25, 2020

Samuel Tracz - 0927451

Checkpoint 2 Documentation

Checkpoint Summary

We implemented the symbol table and type checking for this assignment. The symbol table was a hashmap that contains the name of the variable as the key and an arraylist as the value. The arraylist contains the type and level of the variable name, so that we can verify what scope the variable is used in easily.
We also implemented type checking, which validates that the user is implementing the correct variables in the right places.

Techniques and Design Process

For our symbol table, we created a hashmap that stores the variable name as the key, and it stores an arraylist as the value. Within the arraylist we store the type of the variable and the level of the variable. This means we can easily access the scope and type of the variable when needed. Once we come across a variable we search for the key in the hashmap and then find the most recent element of the arraylist to be used in the scope we are in. Once we leave the scope, the elements of that level/scope are deleted since they cannot be accessed outside of that scope.

For our type checking, we started off checking if things were being assigned to the correct variable type. We then had to check if the variable existed and was usable within the scope, which is the implementation of the symbol table. With strict

type checking, we started with simple assignments such as assigning a number to an int variable and then we worked our way up to operation expressions and function expressions. We then checked if the index of the array that was being accessed was correct. Next, we had to make a check for the parameters and if they were being sent into a function correctly and if they were the correct type. Furthermore, we were made to check a combination of them as well, such as an operation expression inside an index variable.

Lessons Gained

For checkpoint 2, we have learned:

- How to use two data structures inside themselves
    - How to traverse an arraylist inside a hashmap
- How to use variables from different scopes using the symbol table
- What a symbol table is and what it does
- How to check the types of variables in different situations

Assumptions and Limitations

Our assumptions are:

- there can only be int and void variables
- You can't assign a void variable to another void variable
- We assume this is the documentation you are looking for because it doesn't mention it in the assignment

- We assume that we can print out multiple errors for the same thing causing multiple problems
- We assume that we continue with the symbol table once we run into an error
- We assume that the the println and system errors will be printed to different sections, because if you run it as-is, you will get error messages in between the printing of the symbol table
- We assume that the operation expressions will be used appropriately, because you can pass in add(1+1) into a function, but you'll also be able to send in add(1>1) because they are both operation expressions
- We assume there will be a return statement

Our limitations are:

- Well first of all this coronavirus thing made collaboration more limited, with internet problems and technical issues
- (assumption of a limitation) We assume that if you nest enough things, such as an operation expression within an array within a function call for an array index that is used as a parameter for a function call may break the program
- We type check a full array as the same type as a regular variable, so:
    - Int a[10];int b; int newVar = a+b; will work
    - Int a[10];int b; int newVar = a[2]+b; will work

Improvements

We ended up using a global variable for one of the checks because it was a quick fix. We would improve on that part by actually traversing the classes until we found what we were looking for instead of just finding a way around it.

Our code is fairly long and repetitive. If we were to fix it, we would make it into functions, but for our understanding purposes, we ended up copying a lot of code.

Contributions

Samuel

- Documentation
- Pair programming
- Creating test files
- Creating the readme file

Mohammadamin

- Creation of symbol table
- Type checking operations
- Type checking functions and parameters
- Type checking array indexes

Acknowledgement

Prof. Fei Song has provided some of the code for checkpoint 1 in java_tiny, and we have added on to it. Also, we were using the recommended data structure for the symbol table from the chapter 8 slides.