Milica Andric

CS536

P1 part 2

2/6/2021

How to build and invoke the program

For convenience, place all the project files into one folder. This directory should include the following: Sym.java, SymTable.java, DuplicateSymException.java, EmptySymTableException.java, and P1.java. To compile the program, making sure that you are in the directory you just created containing all project files, run the following command:

```
javac *.java
```

To invoke the program, we will be making use of our P1.java file. Run the following command:

```
java P1
```

Test methods

public static boolean testSym() -- Test method for Sym constructor and getType() method.

public static boolean testAddDecl() -- Test method for addDecl method and SymTable constructor.

public static boolean testRemoveScope() -- Test method for addScope() and removeScope() methods.

public static boolean testLookupLocal() -- Test method for addScope(), removeScope(), and lookupLocal() methods.

public static boolean testLookupGlobal() -- Test method for addScope(), removeScope(), addDecl() and lookupGlobal() methods.

public static void testPrint() -- Test method for print() method.

Additional comments

In my previous submission for P1 part 1, I included the below comment without including the 'expected.txt' document mentioned in my zip file. I have included that text document in this submission, under the assumption that my program will be tested against my own P1.java file and not the file provided. Not including this file in my P1 part 1 submission would not affect my grade and the graders would not have been able to run my program because I only provided my P1.java solution anyways, but I just wanted to mention this to clear up any confusion upon initially reading this pdf document with my part 1 submission!

To test functionality of SymTable class' print() method, I have included a file named 'expected.txt'. This file contains the expected output when running the program via the instructions above. Once the program is run, there should be a new file in the current directory named 'output.txt'. Compare these two files to ensure that the expected output is being received. On a Linux machine you can see whether two files match by using the *diff* utility. To compare the two files while ignoring whitespace, run the following:

```
diff -b -B expected.txt output.txt
```