

CS335 Project

Milestone 2

Group ID - 31

Kartavya Damor(200492) Soruabh Mina(200996) Kevalkumar Solanki(200991)

Instructions

The following tools were used in the project :-

- Flex : flex 2.6.4
- Bison : bison (GNU Bison) 3.8.2
- Dot : dot - graphviz version 2.43.0 (0)
- g++ : g++ (Ubuntu 11.3.0-1ubuntu1 22.04) 11.3.0

The MakeFile contains the compiling codes for the flex bison and c++ program,

```
1 compile:
2   bison -dtv parser.y
3   flex scanner.l
4   g++ -o milestone2 parser.tab.c lex.yy.c AST.cpp symbol_table.cpp typecheck.cpp 3ac.cpp
```

Milestone 2:

For Generating Symbol Tables and 3AC :-

- make
- ./milestone2 --input ../tests/<TestFile Name>.java

The output CSVs and TXTs will be generated in a folder (with the same name as testfile).

For your convenience, we have included a bash script, which runs on all our testcases and provides the output in the respective test name folders.

For executing the bash-script,

- chmod u+x runall.sh
- ./runall.sh

Implementation level details

Modifier contains limited keywords (public, private, static)

Since there are C-type array declarations, the function's return value cannot be java style array. For example, "public int[8] fun()" will not work.

To reduce conflicts, we have simplified the grammar to support only basic for loop.

Additional Features Implementations

We have added functionalities for import statements, i.e import statements like import java.util.*

Primitive type cast expressions are supported.

Support for Strings, including a few operations like concatenation, support for printing with println()

References

1. Java Language Specifications: <https://docs.oracle.com/javase/specs/jls/se17/html/jls-19.html>
2. <https://github.com/mohitmo/CS335-Project> (learnt Dot file generation)
3. A.Aho, R.Sethi, and J.Ullman. Compilers: Principles, Techniques, and Tools, 1st edition.
4. Testcases code generated from GFG, javaviz etc websites.