CS335 Project
Milestone 3
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,

```
compile:
  bison -dtv parser.y
  flex scanner.l
  g++ -o milestone3 parser.tab.c lex.yy.c AST.cpp symbol_table.cpp typecheck.cpp 3ac.cpp
```

Milestone 3:
For Generating Symbol Tables and 3AC :-

- make
- ./milestone3 −−input ../tests/<TestFileName>.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
Primitive type cast expressions are supported.
Support for Strings, including a few operations like concatenation, support for printing with println()

3AC Functions

1. param: Pushes the parameters into the stack from right to left.
2. stackpointer– x : Removes x bytes from the stack
3. popreturn: returns the function's return value from its corresponding location (offset) stack.
4. popparam: Emits parameters from the stack in the correct order. (First emitted value will be the base pointer)

5. beginfunc x : Allocates x bytes (for local and temporary variables) in the stack for the called function.
6. call func x : Calls the function (whose x parameters were previously pushed onto the stack.)
7. call_alloc x : Allocates x memory for the object and return the object.

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.

:)