Compilers: Milestone 1

How to Run

- cd $< milestone >$
- make
- ./myASTGenerator $--$input $./tests/ < test\_input$ (.java format)$>$ $--$output $< output\_file$ (.dot format)$>$ $--$verbose (True/False)

Description

We have used ANTLR tool to parse the Java input. ANTLR provides a parse tree representation of the code. Further transforming it, we built an Abstract Syntax Tree (AST) by DFS traversal of parse tree.

The parser is fully accustomed to handle basic Java functionalities (required in the milestone 1). Furthermore, the parser is built in with additional functionalities:

- Support for Interfaces
- Static polymorphism via method overloading
- Dynamic polymorphism via method overriding
- Type casting
- Generics

Flags Description:

- $--$ input : input file name, (*default = inputfile.java*)
- $--$ output : input file name, (*default = new.dot*)
- $--$ verbose : prints the output of the input on the screen
- $--$ help : help regarding flags

Graphviz tool has been used for graphical representation of AST