# JFlex Software Design Document

Nikaylah Woody April 24, 2018

# 1. Overview

This software provides an application of a compiler in Java using the Pascal language as described in the grammar found in the design below. It will also convert the code to MIPS.

# 2. Design

```
Our small Pascal keywords and symbols,
the definitive list.
    KEYWORDS                SYMBOLS
    ________                _______
 1) and                     ;
 2) array                   ,
 3) begin                   .
 4) div                     :
 5) do                      [
 6) else                    ]
 7) end                     (
 8) function                )
 9) if                      +
10) integer                 -
11) mod                     =
12) not                     <>
13) of                      <
```

14) or<br>
15) procedure<br>
16) program<br>
17) real<br>
18) then<br>
19) var<br>
20) while

<=<br>
><br>
>=<br>
*<br>
/<br>
:=

# 3. Implementation

The **scanner** will scan through each symbol and number to see to make sure it is indeed a part of the grammar.

The **parser** will recognize whether an input string of tokens is a valid Mini- Pascal program as defined in the grammar. I has a Token object that contains String lexeme and TokenType

The **symbol table** will store information about identifiers found in the Pascal program. It will be integrated with the recognizer.

The **syntax tree** will contain a tree that contains nodes.

The **semantic analyzer** will be implemented in three different way in an independent module. It uses the syntax tree and check to see if the following works otherwise and error arrives:

-It will make sure all variables are declared before they are used

-It will assign a type, integer type or real type, to each expression. And the type of any expression node in the tree should be printed in the tree’s indentedToString.

-It will also make sure that types match across assignment.

The **Code Generation** module will take the syntax tree as it’s input, go through the semantic analyzer and return a string with the MIPS assembly language code as its output

# 4. Compiler- Final version

The final version of this project will include the Jar file which will contain a .jar executable in a new product folder. Along side the User Manual to use the final product.