**Scanner Documentation**

FLCD – Lab1

**Scanner implementation:**

- Identifiers of maximum 250 characters

- Separate tables for constants and identifiers

- Lexicographically sorted table

**Scanner specification:**

The scanner has the following input files:

- “program.txt” - source code

- “operators.txt” - operators with their specific code, separated by space, one on each line

- “separators.txt” - separators with their specific code, separated by space, one on each line

- “reserved\_words.txt” - reserved words with their specific code, separated by space, one on each line

The scanner outputs the files:

- “pif.txt” - the program internal form, one on each line

- “identifiers.txt” - identifiers symbol table

- “constants.txt” - constants symbol table

- “error.txt” - lexical errors, with their specific line

**How the scanner works:**

The scanner reads all the input files and remembers the reserved words, operators and separators. Then, it parses the source file line by line, applying the following changes: replaces tabs with space (“ “), adds space (“ “) before and after separators and operators, deletes excess spaces then splits the line in tokens (using space as a delimiter).

Each token is then analyzed. If it's a separator, operator or a reserved word, it's written into the pif file. If not, the scanner checks if the token is in the identifier symbol table or the constants symbol table and writes them to the pif file. If they are not, then the functions isValidIdentifier(String symbol) and isValidConstant(String symbol) check to see if they respect the rules of the language and an error is written in the error text file if they don't.

**Main Classes:**

Analyzer: Responsible for most of the work.

Methods:

- replaceTabsWithSpaces(String code);

- organizeLineWithSpaces(String startingLine): adds spaces before and after separators and operators

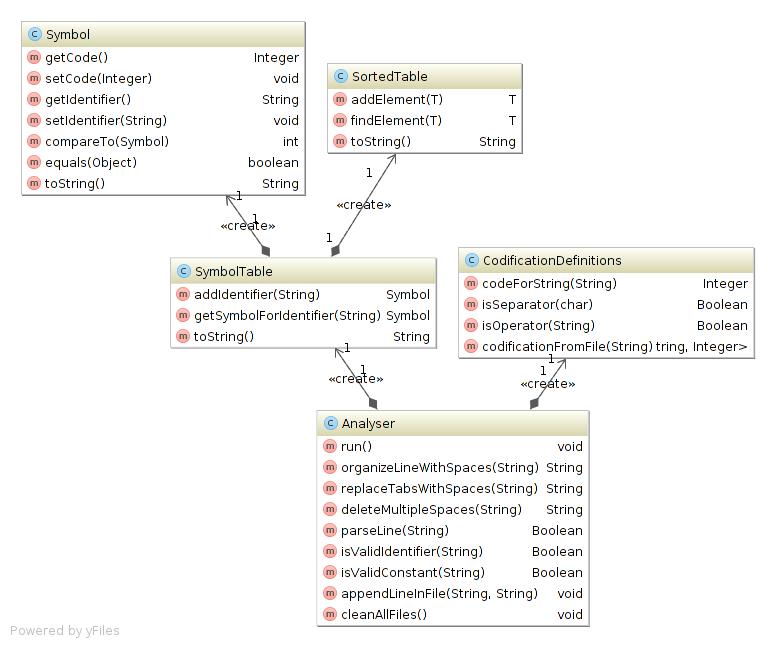
- deleteMultipleSpaces(String line): deletes excess spaces

- parseLine(String line): splits the line into multiple tokens and analyses them, then adds corresponding information to each output file

- isValidIdentifier(String symbol)

- isValidConstant(String symbol)

**Class Diagram:**

****