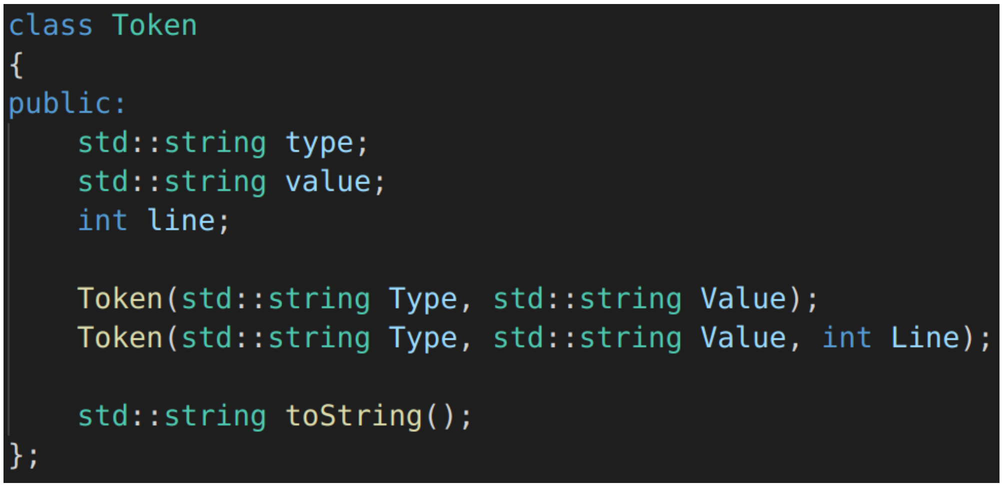
Chis Sergiu
group 932/1

Lexical Analyzer documentation

Github: https://github.com/SergiuChis/FLCD

The lexical analyzer makes use of a data type called "Token". Token has 3 fields: a type, a value, and the line where it sits in the code.

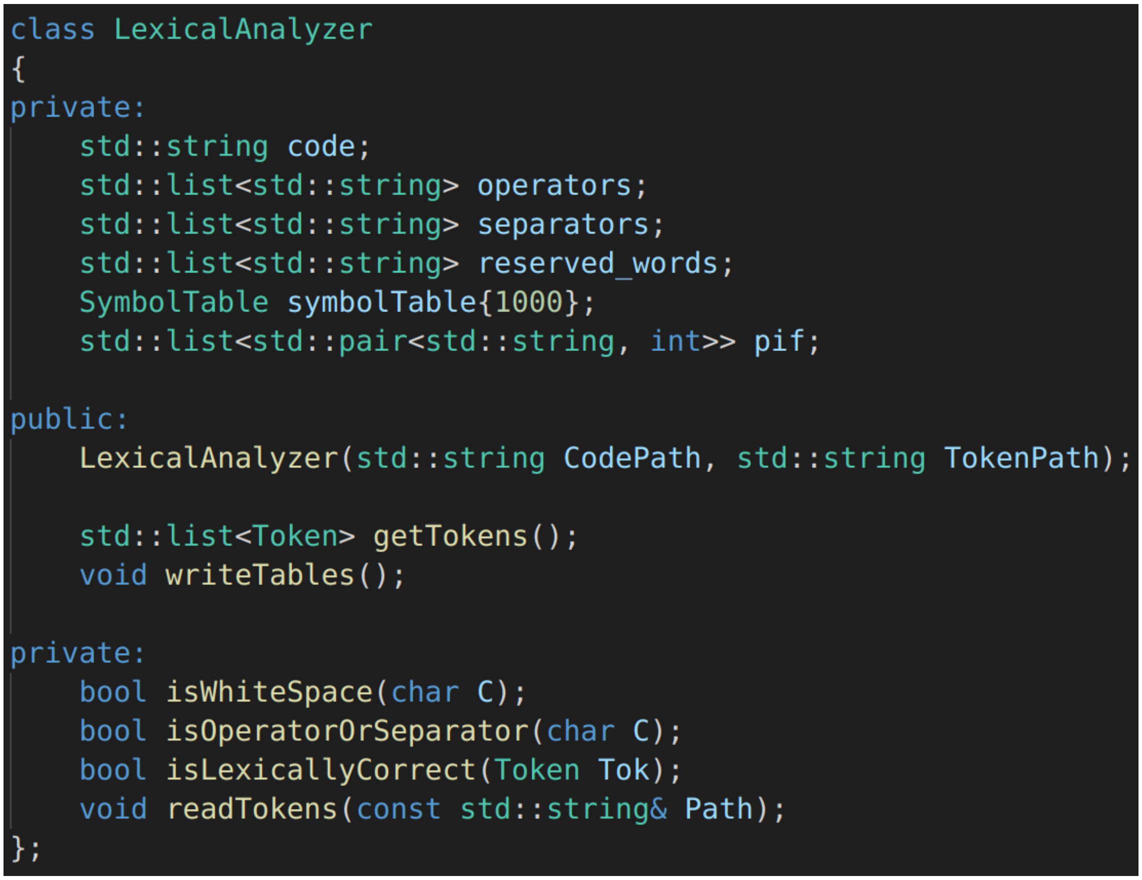


```
class Token
{
public:
    std::string type;
    std::string value;
    int line;

    Token(std::string Type, std::string Value);
    Token(std::string Type, std::string Value, int Line);

    std::string toString();
};
```

The main class that handles the tokenization and the lexical analysis, is represented as follows:



```
class LexicalAnalyzer
{
private:
    std::string code;
    std::list<std::string> operators;
    std::list<std::string> separators;
    std::list<std::string> reserved_words;
    SymbolTable symbolTable{1000};
    std::list<std::pair<std::string, int>> pif;

public:
    LexicalAnalyzer(std::string CodePath, std::string TokenPath);

    std::list<Token> getTokens();
    void writeTables();

private:
    bool isWhiteSpace(char C);
    bool isOperatorOrSeparator(char C);
    bool isLexicallyCorrect(Token Tok);
    void readTokens(const std::string& Path);
};
```

- class attributes:
    - code (the source code as string)
    - operators (the list of all operators defined in the language)
    - separators (the list of all separators defined in the language)
    - reserved_words (the list of all reserved words defined in the language)
    - symbolTable (the symbol table of the input program)
    - pif (the PIF of the input program)

- class methods:
  - getTokens();
    - this method parses the source code, splits it into tokens, populates the symbol table and the PIF, and outputs a message telling if the program is lexically correct or there is a lexical error
    - return value: the list of tokens found

  - writeTables();
    - this method writes the symbol table and the PIF to two different files (ST.out and PIF.out)

  - isWhiteSpace(char C);
    - this method checks if a character is one of the following: space, tab, or newline
    - return value: true if it is space, tab, or newline

  - isOperatorOrSeparator(char C);
    - this method checks if a character is an operator or a separator
    - return value: true if it is an operator or a separator

  - isLexicallyCorrect(Token Tok);
    - this method checks if a token is lexically correct
    - return value: true if the token is lexically correct

  - readTokens();
    - this method reads all the predefined tokens from the input file, and populates the appropriate lists