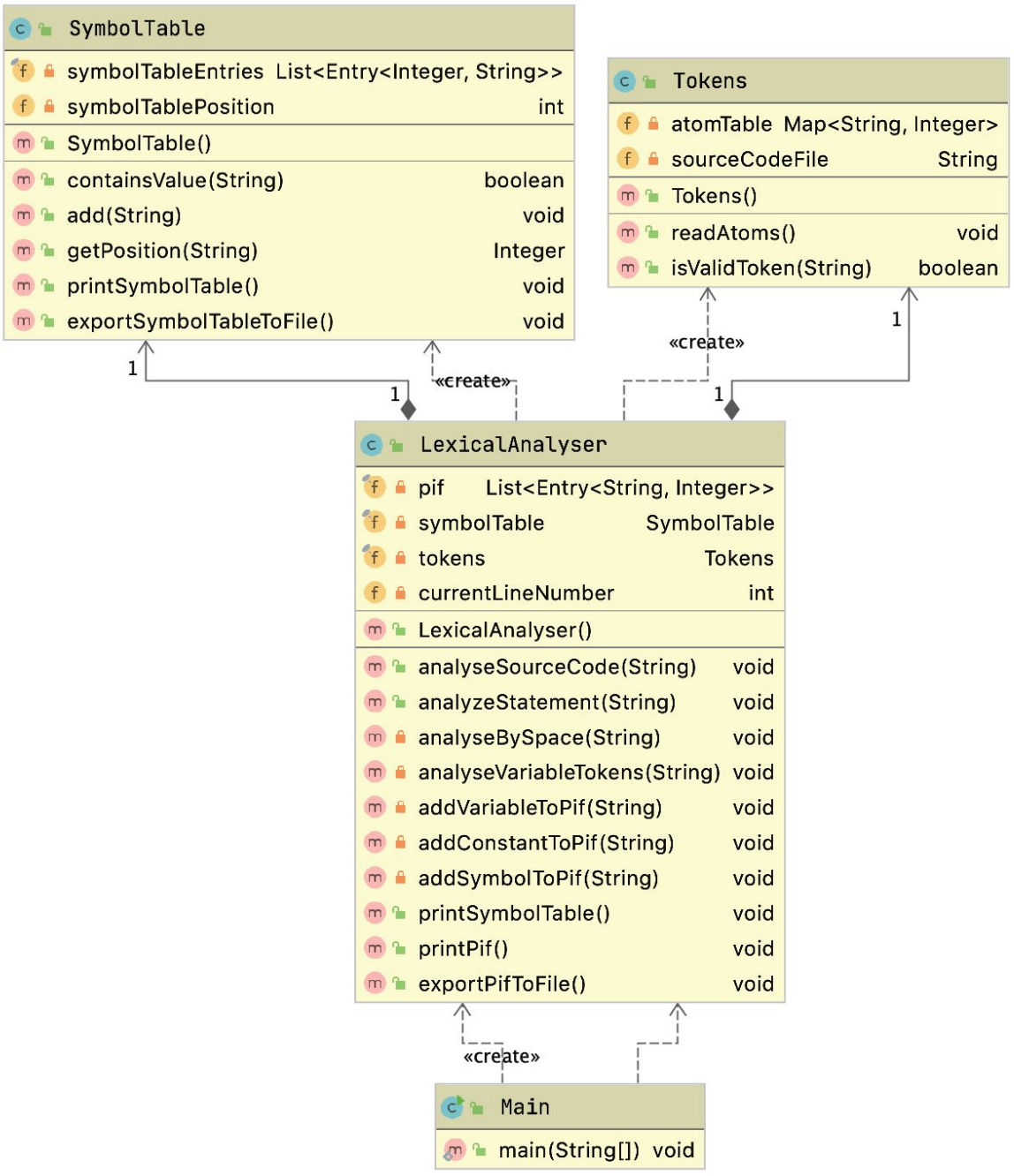
Limbaje formale si tehnici de compilare - Lab Work for lab3 (Documentation)

Statement: Implement a scanner (lexical analyzer): Implement the scanning algorithm and use ST from lab2 for the symbol table.

(JavaDoc available in the source code)

Class diagram:



Class SymbolTable

Documentation on methods available for Lab2

Class Tokens

public void readAtoms() -> Read atoms from tokens.in

Class LexicalAnalyser

package services;

```
    /**
     * Reads the source code file, line by line, and analyses each statement.
     *
     * @param fileName name of the source code file
     */
    public void analyseSourceCode(String fileName)


    /**
     * Method for analyzing a statement, by first splitting it after = to analyse the
assignments and new declarations
     * and then after space, to verify whether symbol - variable or constant - or
reserved word. Before splitting for =,
     * the statement is verified for == and != to analyse the conditions (so that == and
!= is not interpreted as an
     * assignment or declaration).
     *
     * @param statement line statement to be analysed
     */
    public void analyzeStatement(String statement)


    /**
     * Splits the input value by space and analyses each value item, adding it
correspondingly to PIF and the ST
     *
     * @param value to be analysed
     */
    private void analyseBySpace(String value)
```

```
    /**
     * Analyse input variable to verify whether it is a new declaration or assignment
and populate the st and pif with it.
     *
     * @param variable variable to be analysed
     */
    private void analyseVariableTokens(String variable)


    /**
     * Adds a variable, if not empty, to the PIF.
     *
     * @param valueItem variable to be added to PIF
     */
    private void addVariableToPif(String valueItem)


    /**
     * Adds a constant, if not empty, to the PIF.
     *
     * @param valueItem constant to be added to PIF
     */
    private void addConstantToPif(String valueItem)


    /**
     * Prints the PIF
     */
    public void printPif()



    /**
     * Exports the PIF to File
     */
    public void exportPifToFile()
```