**Scanner documentation**

1. **Implementation**

The scanner input will be a text file containing the source program, and will produce as output the following:

- PIF - Program Internal Form

- ST - Symbol Table

In addition, the program should be able to determine the lexical errors, specifying the location, and, if possible, the type of the error.

The scanner assignment will be differentiated based on:

1. Identifiers: a. length at most 8 characters
2. Symbol Table: a. unique for identifiers and constants
3. Symbol Table Organization: c. hashing table
4. **Technologies:**
   * Language: Python
   * IDE: PyCharm
5. **Data Structures**
   * Symbol Table: HashTable – having each element represented by a tuple (variable, unique\_code).
   * Program Internal Form – a list of tuples (code, id)
   * code the value from its corresponding key in the codification table for separators/operators/reserved words and for identifiers and constants it is the second value from its corresponding tuple in the symbol table representing a unique code.
   * Id: -1 for separators/operators/reserved words, 0 for identifier or 1 for constant.
   * The special symbols as separators, operators and reserved words are represented in a list.
   * The codification table is a dictionary in which the key is either a separator/operator/reserved word or an identifier/constant and the value is a natural number for the first category greater than 1 and for identifier is 0 and for constant is 1.
6. **Flow**

The program is read from a txt file, line by line and split up into tokens using the mthod detectToken. If the given token is an operator or a separator or a reserved word I added to the PIF with code -1. Next we check if our token is an identifier or a constant and in both cases we add in the ST and with the id code generated we add in the PIF with its corresponding identifier.constant code which is either 0 or 1 and with the id from the ST.