https://github.com/andrei47w/LFTC-ubb/tree/master/lab3

Scanner steps:

- splits the file by lines

- parses through each character from the line and does 3 checks:

- if the token is an operator (=, <, >, etc.)

- if the token is a string (is surrounded by “)

- if the token is a separator (;, {, [, etc.)

- else it’s an identifier

- if the result is a known token we add it in PIF with the ID (-1, -1)

- else we use the ID from the symbol table in PIF

- if it isn’t a token or cannot be stored in the symbol table, it is an error

Input: token.in, program.txt

Output**:** PIF.out, ST.out, message “lexically correct” or “lexical error + location”

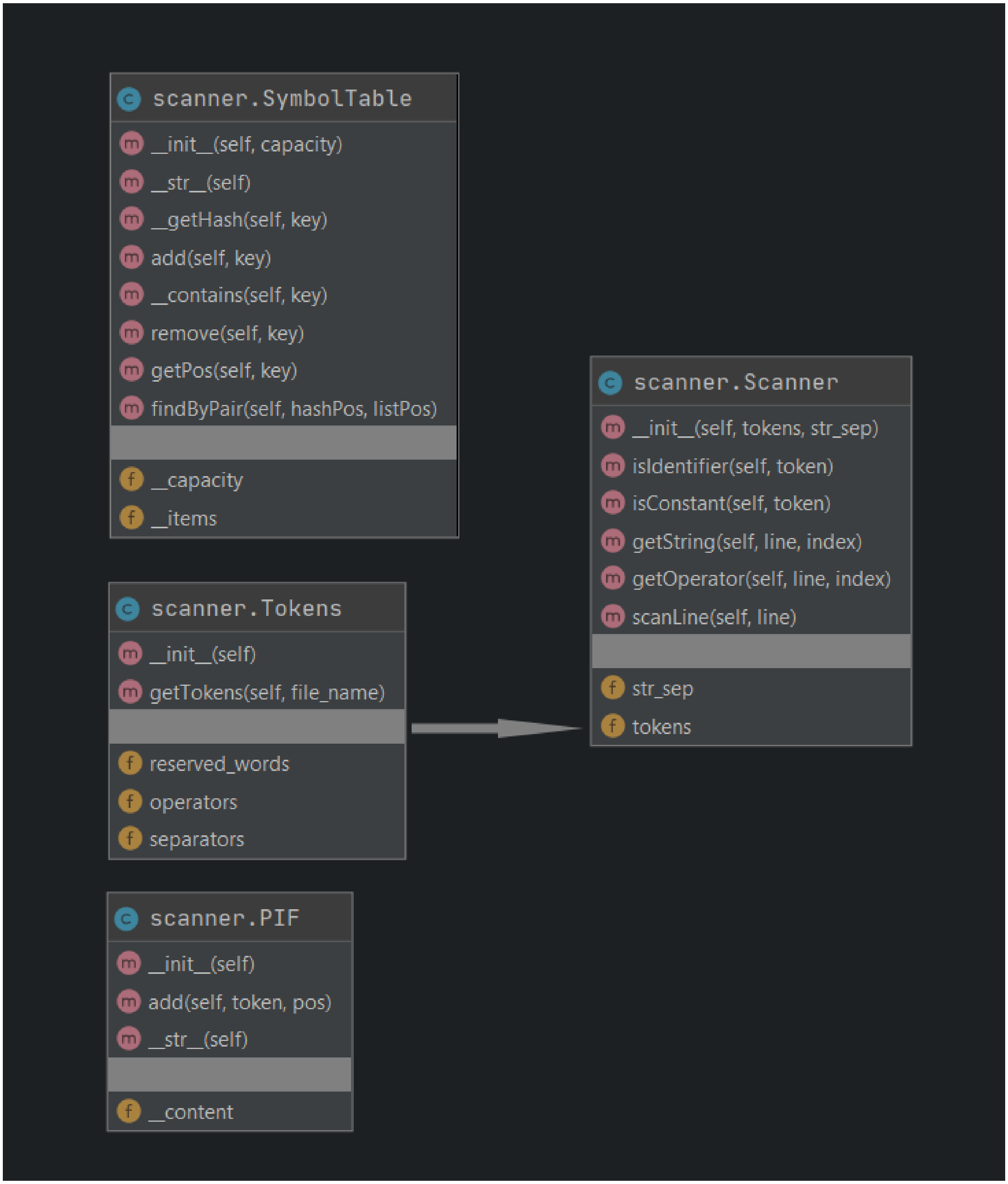
Regex uses:

Identifier: ^[a-z]([a-zA-Z]|[0-9])*

-must begin with a lowercase letter and can continue with 0 or more letters or digits

Constant: ^(0|[+\-]?[1-9][0-9]*)$

-must begin with a 0, + or – and must start with a non-zero digit followed by 0 or more digits


scanner.SymbolTable
__init__(self, capacity)
__str__(self)
__getHash(self, key)
add(self, key)
__contains(self, key)
remove(self, key)
getPos(self, key)
findByPair(self, hashPos, listPos)
__capacity
__items
scanner.Scanner
__init__(self, tokens, str_sep)
isIdentifier(self, token)
isConstant(self, token)
getString(self, line, index)
getOperator(self, line, index)
scanLine(self, line)
str_sep
tokens
scanner.Tokens
__init__(self)
getTokens(self, file_name)
reserved_words
operators
separators
scanner.PIF
__init__(self)
add(self, token, pos)
__str__(self)
__content