https://github.com/CronosClaus/flcd2/tree/main/flcd/lab4

The class Scanner has:

* self.__reserved_word
* self.__operator
* self.__identifier
* self.__constant
* self.__separator

that are regex strings for matching their respective token

* self.__tokens =

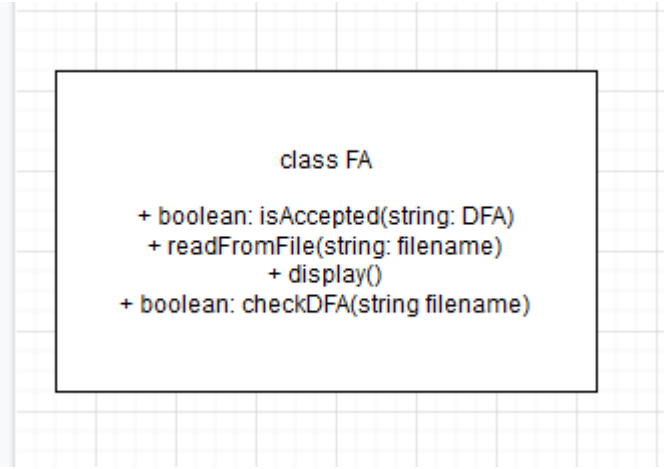
[] list of tokens

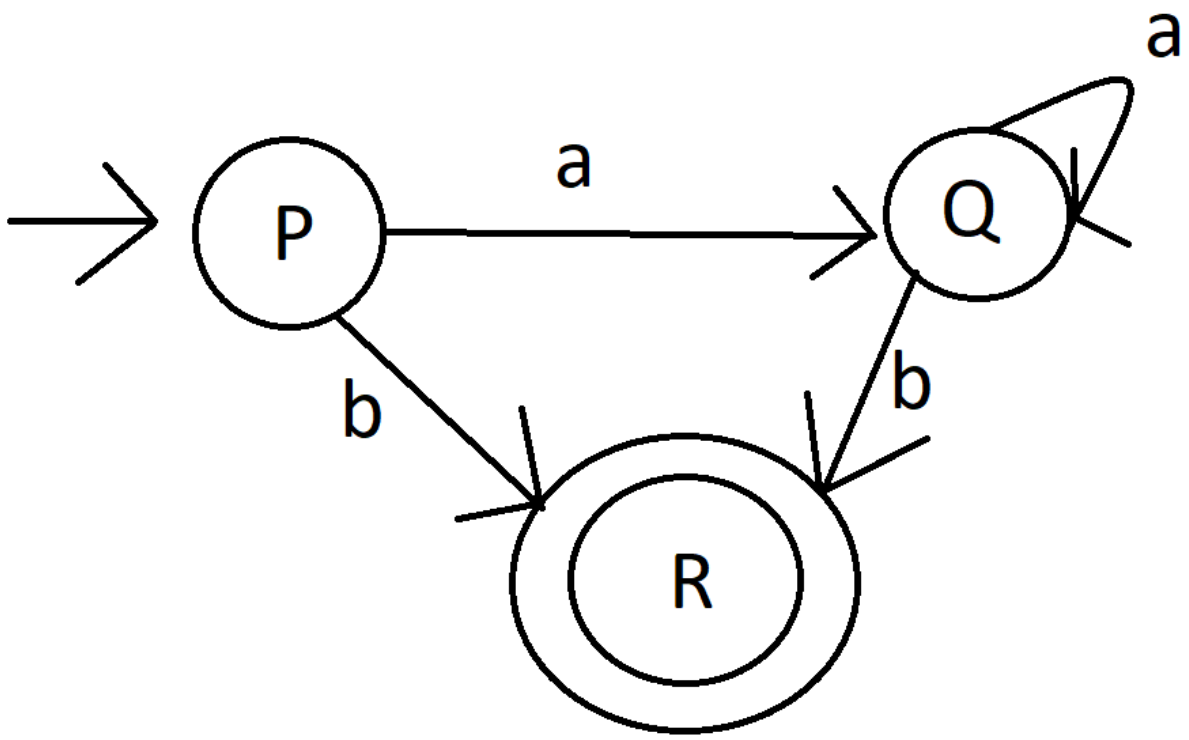
* self.__pif = []
* self.__sti = SymbolTable()
* self.__stc = SymbolTable()

pif is the program internal form a list of tuples (token, number)
if token is reserved word or operator or separator number is -1
else is the position in which you find the token in ST

sti and stc are the simbol tables for identifiers and constants

respectively the only method of this class is
* def scan(filename)
that opens the file reads line by line and tokenizes lines into atoms, then for each atom
it tries to find it's type and if it doesn't find it then it prints on stdout the lexical error
and if there's no lexical error then it prints "Lexically correct"





EBNF form of FA.in

```
file = {line}
line = state '=' '{' state ':' alphanum {',' state ':' alphanum} '}'
state = ['*']alpha['+']
alphanum = alpha | num
alpha = 'a' | 'b' | ... | 'z'
num = '0' | '1' | '2' | ... | '9'
```