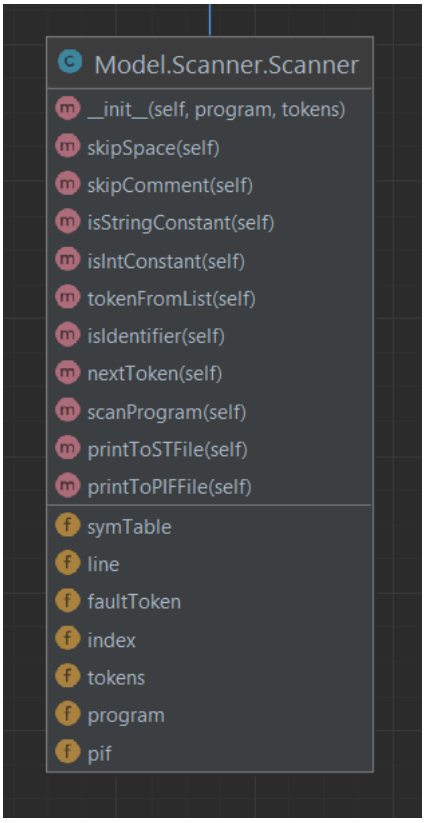
Scanner documentation

https://github.com/MirceaDragosVlad917/FLCD/tree/main/Lab3

Statement: Implement a scanner(lexical analyzer): Implement the scanning algorithm and use ST from Lab 2 for the symbol table.
Input: Programs p1/p2/p3/p1err and token.in(see Lab 1a)
Output: PIF.out, ST.out, message “Lexically correct” or “Lexical error + location”
Deliverables: input, output, source code, documentation
Details:

- ST.out should give information about the data structure used in representation
- If there is an error, the program should give a description and the location (line and token)

Class diagram

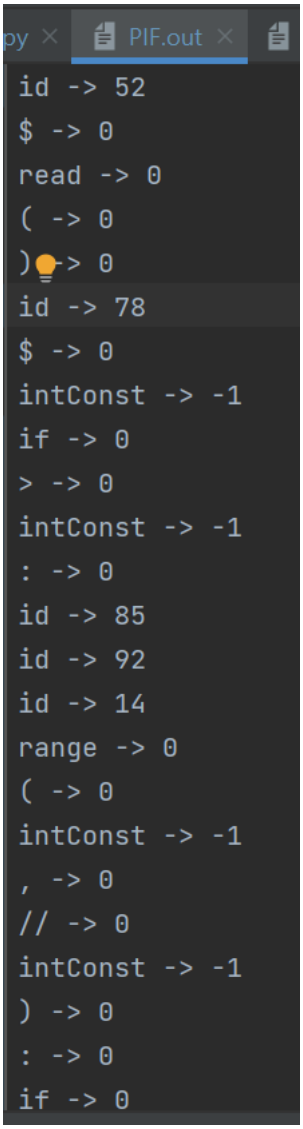


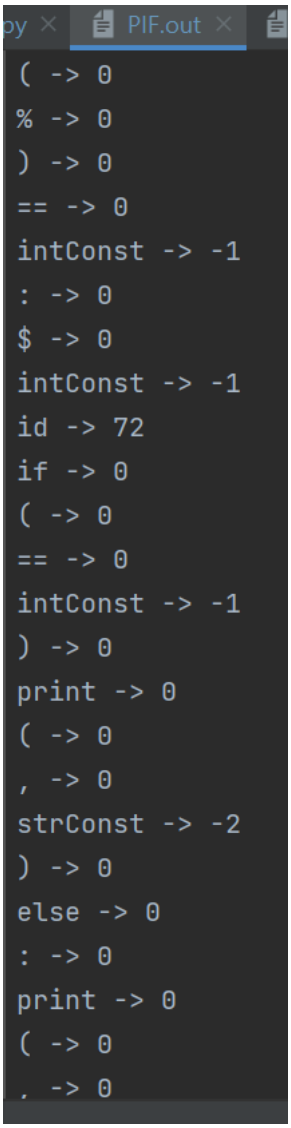
The implementation uses regexes to to find matches with parts of the program to string constants, int constants, identifiers or tokens. The scanner reads the program as a string from a file and the tokens list from token.in. The function scan performs a parsing of the program by going element by element, skipping white spaces and comments, matching each

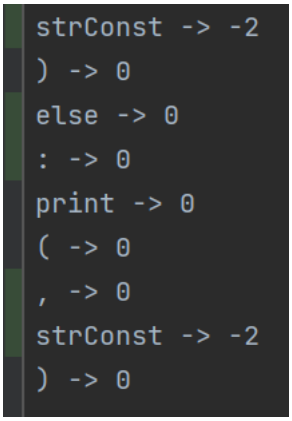
element to the corresponding field and adding them both to the pif, in the pif.out file, and the symbol table, in the ST.out file. It also raises errors in case some elements can not be classified. When an element is adding to the pif, identifiers correspond to their position in the symbol table, string constants correspond to the value -2, int constants to the value -1 and tokens to the value 0.

Tests

This is the PIF.out file for problem p1





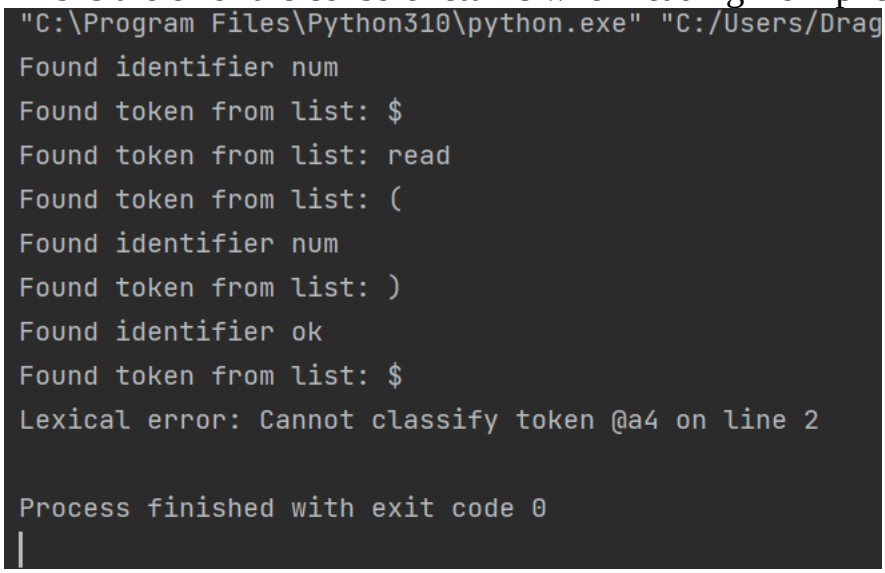


P1:

```
num $ read(num)
ok $ 0
if num > 1:
   for i in range(2, num // 2):
      if (num % i) == 0:
```

```
        ok $ 1
        break
  if (ok == 0)
    print(num, "is a prime number")
  else:
    print(num, "is not a prime number")
else:
print(num, "is not a prime number")
```

This is the error the console returns when reading from p1err



P1err:

```
num $ read(num)
ok $ @a4
if num > 1:
  for i in range(2, num//2):
    if (num % i) == 0:
      ok $ 1
      break
  if (ok == 0)
    print(num, "is a prime number)
  else:
    print(num, "is not a prime number")
else:
print(num, "is not a prime number")
```