Scanner - documentation

The PIF

The Program Internal Form (PIF) is an array of pairs. These pairs can be of two forms:

- <token, -1> (where ‘token’ is a valid token of the programming language)
- <id, st_position> (where ‘id’ is a placeholder for a valid identifier (or constant) in the programming language, which sits at position ‘st_position’ in the symbol table)

For the implementation, I used a simple Python list, which holds tuples of type (string, string). Additionally, I used two symbol tables: one for the identifiers and the other for constants. The PIF objects will be managed by the Scanner class.

The Scanner

The Scanner class contains a PIF and two ST objects.

The scanning algorithm is an implementation of the pseudocode algorithm from the second course.

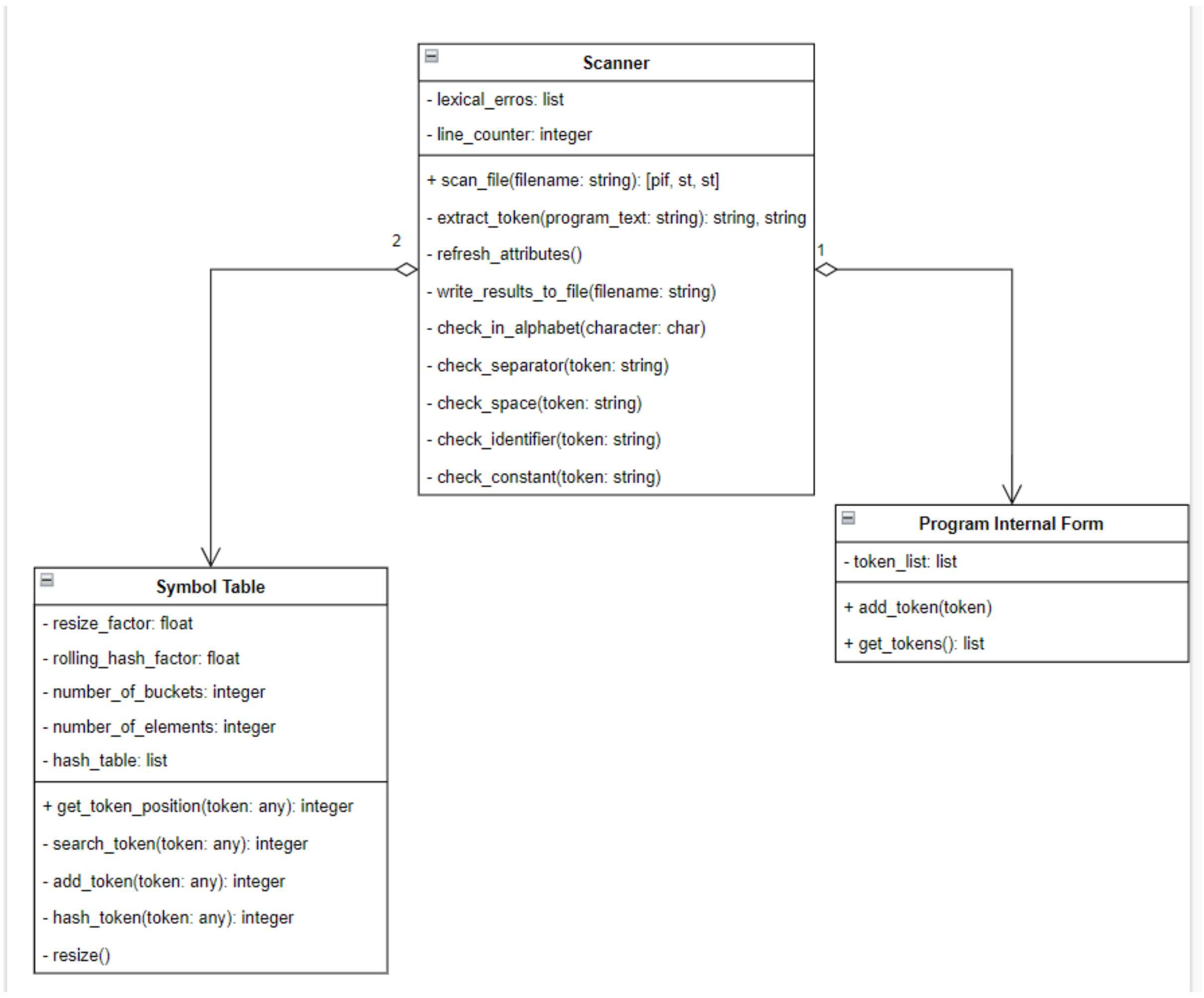
In short, the scanning splits a given source code into tokens, which are added to the PIF. If a token is a valid token (reserved word, separator, operator) of the programming language, it is added with a ‘-1’ value in the PIF (however, spaces are omitted to save memory). Otherwise, if the token is an identifier or constant, it is added to its respective symbol table, and its position in the ST will be added to the PIF (together with the identifier ‘id’ or ‘const’). If a token is invalid, the scanning algorithm should signal an error.

The scan results are stored in ‘pif’ and ‘st’ output files (as text tables), and messages are printed if the source code is correct or has errors (and specifies the line and token with an error).

For checking if a token belongs to a class, I used regular expressions:

- a character is in the alphabet: `[a-zA-Z0-9/*\-+(){}\[\];\'\"!<=>_ \n\t]`
- a token is an identifier: `[a-zA-Z][a-zA-Z0-9_]*`
- a token is an integer constant: `0|([+-]?[1-9][0-9]*)`
- a token is a float constant: `0|([+-]?[1-9][0-9]*)(\.([0-9]+))?`
- a token is a bool constant: `(TRUE)|(FALSE)`
- a token is a character constant: `'[a-zA-Z0-9_+\-*/:]'`
- a token is a string constant: `"[a-zA-Z0-9_+\-*/:]*"`

The class diagram:


Scanner
- lexical_erros: list
- line_counter: integer
+ scan_file(filename: string): [pif, st, st]
- extract_token(program_text: string): string, string
- refresh_attributes()
- write_results_to_file(filename: string)
- check_in_alphabet(character: char)
- check_separator(token: string)
- check_space(token: string)
- check_identifier(token: string)
- check_constant(token: string)
2
1
Symbol Table
- resize_factor: float
- rolling_hash_factor: float
- number_of_buckets: integer
- number_of_elements: integer
- hash_table: list
+ get_token_position(token: any): integer
- search_token(token: any): integer
- add_token(token: any): integer
- hash_token(token: any): integer
- resize()
Program Internal Form
- token_list: list
+ add_token(token)
+ get_tokens(): list