https://github.com/mircea-negrau/University-Lab-FLCD

The Scanner class contains two instances of the SymbolTable (one for constants and one for identifiers) as well as an instance of a ProgramInternalForm.

On initialization, the Scanner loads from a file the tokens for operators, separators and keywords.

When scan method is triggered, the Scanner loads from a file its contents (the code to be lexically scanned).

It first loads the contents into memory line by line (vector of strings).
Then, it goes line by line and separates the tokens which are not to be taken together (e.g., the ";" gets separated by a whitespace). This part makes sure the multi-symbol tokens are handled properly (i.e. "!=" does not get split into "! =" – notice the whitespace).

Then, the scanner goes line by line and token by token and tries to identify the type of the token (constant, identifier, keyword, separator, operator or unknown).
If the token type is `Identifier`, the token gets added to the identifiers symbol table and to the PIF.
If the token type is `Constant`, the token gets added to the constants symbol table and to the PIF.
If the token type is `Unknown`, a LexicError gets thrown and the scanner stops.
Otherwise, the token gets added to the PIF.

For identifying the tokens, the Scanner uses the following methods:

- isSeparator: searches for the token in the list of separators fetched from the token.in file
- isKeyword: searches for the token in the list of keywords fetched from the token.in file
- isOperator: searches for the token in the list of operators fetched from the token.in file
- isIdentifier: matches the token to regex `^[a-zA-Z]+[a-z|A-z|0-9]*$`
- isConstant: matches the token to the regex
    - `[True|False]|[^0$]|[^[1-9]+[0-9]*$]|[ ^\"[a-z|A-Z|0-9|_]+\"$]`