Title: Lexical Analyzer for a Simple Programming Language

Description:

The LexicalAnalyzer class in this script provides functionality for performing lexical analysis on a simple programming language. It identifies and processes different types of tokens, including reserved keywords, special symbols, constants, and identifiers, and stores the results in the Program Internal Form (PIF). The class uses a SymbolTable object to manage identifiers and their positions.

Components:

- LexicalAnalyzer: A class that performs lexical analysis on input text, identifying and processing various tokens.

- SymbolTable: A class that manages identifiers and their positions using a hash table data structure.

Methods:

- lexical_analysis(): Performs lexical analysis on the input text. It tokenizes the text and checks each token against predefined patterns for reserved keywords, special symbols, constants, and identifiers. It utilizes regular expressions for pattern matching and handles unrecognized tokens as lexical errors. The method updates the PIF and SymbolTable accordingly and outputs the results to ST.out and PIF.out files.

Attributes:

- __symbol_table: An instance of the SymbolTable class that manages identifiers and their positions.

- __pif: A list that stores the tokens and their corresponding positions in the Program Internal Form.

- __reserved_keywords: A set containing reserved keywords recognized by the lexical analyzer.

- __special_symbols: A set containing special symbols recognized by the lexical analyzer.

Usage:

1. Initialize an instance of the LexicalAnalyzer class with a SymbolTable object.

2. Call the lexical_analysis() method with the input text to perform lexical analysis.

3. View the output in the ST.out and PIF.out files, which contain the SymbolTable and the Program Internal Form, respectively.

4. Check the console for any lexical error messages or confirm the correctness of the lexical analysis.

File Output:

- ST.out: Contains the SymbolTable, which manages identifiers and their positions.

- PIF.out: Contains the Program Internal Form, which stores the tokens and their corresponding positions.

Note:

Ensure that the SymbolTable class and any required external modules are accessible and correctly implemented to enable seamless execution of the LexicalAnalyzer class.