LEXER ANALYZER

Q2: Give two functionalities of the project?

Answer:

Control Flow Explanation:

1. Input and Initialization:

The Result method is the entry point.

Input text is provided for lexical analysis (txt), along with an optional file path for transition rules (tt).

Initialization of variables (txtIndex, iState, cTemp, cChar, sToken, flag).

2. Transition Rules Loading:

loadTransitionTable is called within Result to load transition rules.

If successful, the rules are stored in the rules list.

3. Lexical Analysis Loop:

Loop Initialization:

Sets up a loop to iterate through each character in the input text.

Character Processing:

Character Analysis:

Checks for comments, handles one character at a time.

Uses getNextState to determine the character type and transition between states.

Tokenization and Output:

Constructs tokens based on transitions.

Identifies and categorizes tokens (identifiers, keywords, numbers, symbols).

Builds the output string by appending identified tokens or placeholders.

4. Functions Involved in Control Flow:

loadTransitionTable Function:

Reads transition rules and initializes the rules list.

getNextState Function:

Determines the next state based on the current state and input character.

Guides the transition between different states in the lexical analysis process.

isKeyword Function:

Determines if a given token is a keyword by comparing it to a predefined list of keywords.

5. Error Handling:

Exception Handling:

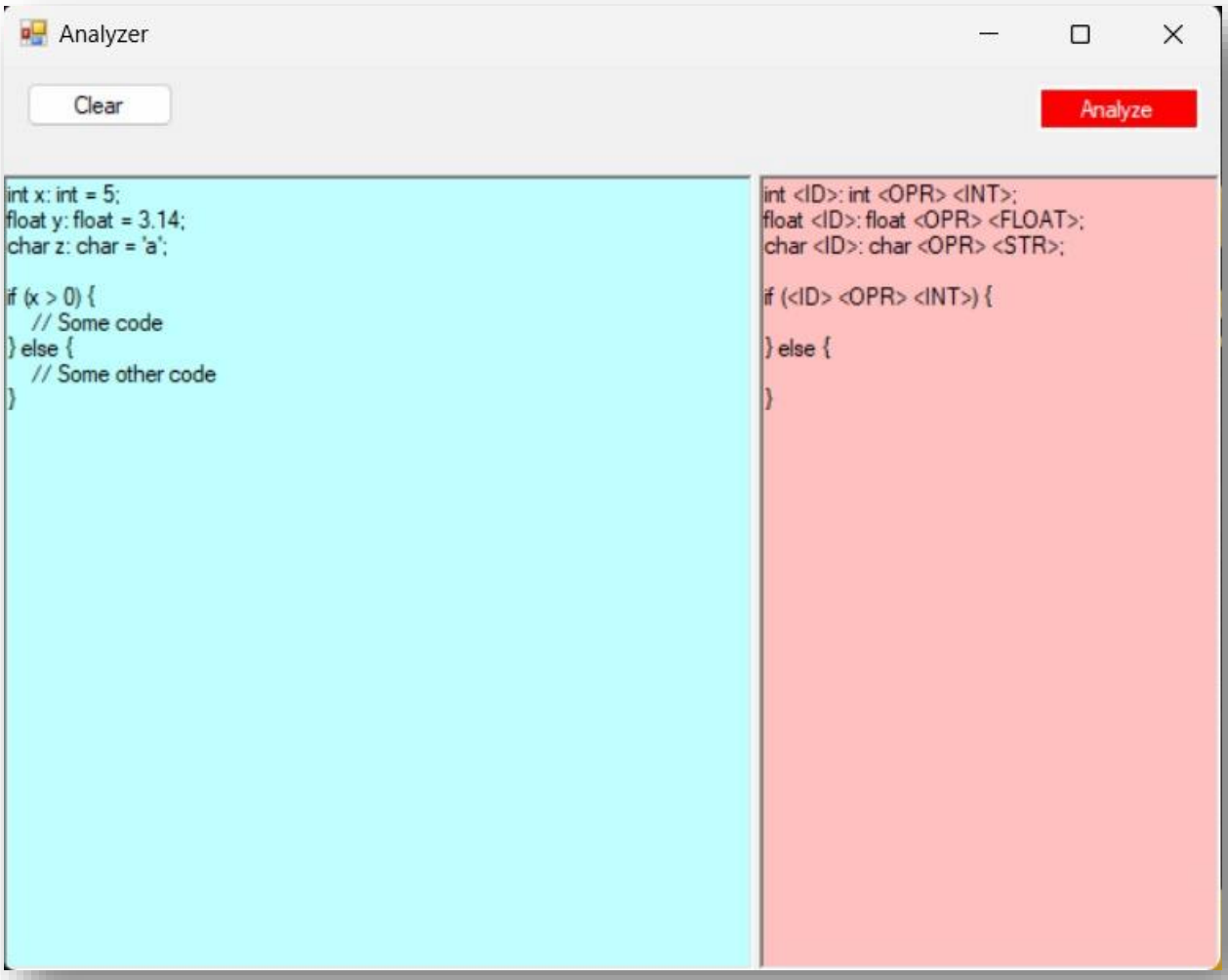
If there's an issue loading transition rules, an exception is caught.

Returns an error message indicating the failure to load transition rules.

6. Output:

Result String:

The output of the Result method is the processed string based on identified tokens and placeholders.


Analyzer
Clear
Analyze
int x: int = 5;
float y: float = 3.14;
char z: char = 'a';
if (x > 0) {
// Some code
} else {
// Some other code
}
int <ID>: int <OPR> <INT>;
float <ID>: float <OPR> <FLOAT>;
char <ID>: char <OPR> <STR>;
if (<ID> <OPR> <INT>) {
} else {
}