A brief explanation of how the code works is as follows:

* The program is a top-down recursive-descent parser that also evaluates expressions as it parses.
* It reads a file containing multiple let ... in ... end; blocks, parses each block in turn, and prints the result or Error.
* The code first converts the raw text into a stream of tokens.
* The Lexer class reads characters, groups them into meaningful tokens.
* It also recognizes whether a numeric constant should be int or real and distinguishes keywords from identifiers.
* The Parser class then consumes tokens from the Lexer and applies the grammar rules.
* If anything, unexpected happens it sets an error flag and prints "Error" for that block.
* As the parser descends through the grammar, it evaluates expressions immediately.
* If the parser hits a mismatch it sets error flag = True.