

Requirements:

- Implement a simple interpretive language (GIT language) that can do arithmetic expressions with variables.
- Read the expressions from a given file. For example, if a file named program.git contains
- Operators: +, -, *, /, = have the same meaning and same precedence as in Java.
- • Uniary operators sin, cos, sqrt, log, abs, tan, exp have the same meaning as in
- Java Math functions. These operators take one double and return a double as the result. For example sin 90 returns 1.0.
- Other uniary operators have different meanings:
 - ◦ Operator var defines a double variable in memory, returns 0.0
 - ◦ operator print prints the given parameter on the screen and returns the printed value
 - ◦ operator input asks the user to enter a value from keyboard and assigns that value to the parameter variable. Also returns the assigned value.
- Uniary operators have higher precedence than the binary operators.
- Variables can be regular Java variable names other than reserved uniary operator names
- If there is an error in the expressions then you should print it clearly such as
 - ◦ Undefined variables
 - ◦ Uninitialized variables
 - ◦ Unknown operators
 - ◦ Unmatched parenthesis, etc
 - ◦ No Lvalue, etc

Analysis and Design:

We have a Operand class, and Executable interface. All tokens (except operators) will be in a Operand class that returns its value. Commands are also implements Executable class that to do operations

For example;

- print, input, var commands and functions(sin, cos etc) each have a class extends class of Operator and implements Executable class. They will execute some operations with methods of Executable interface, then return their value by method of ReturnValue() by Operand class.
- Variables have a class called Variable, keeps variable name and its value. This class also extends Operand