INTERPRETER

Project Documentation

1 Interpreter

Interpreter is a computer program that directly executes, i.e. performs, instructions written in a programming or scripting language, without previously compiling them into a machine language program.

In the project we are implementing the interpreter similar to that of the Python interpreter that does some of the basic Arithmetic operations (Addition, Subtraction, Multiplication, Division and Negation) where the input is given in the form of expressions involving numerics or variables.

The interepreter code can work for both single input (A string containing the expression to be evaluated is given in the inputType variable in main function of the Interpereter.py file) and continuous interactive input from terminal.

Continuous interactive input from terminal - example

```
>>>3+4
7
>>>a = 34
>>b = 2
>>>a * b
68
>>> (2+4)-(100-50)
-44
```

2 Execution of Source code

1. Navigate to "Source_Code" folder.
2. For single input, the inputType variable in main function of the Interpereter.py file should be set to a string containing the expression to be evaluated.
3. For continuous interactive input from terminal, the inputType variable in main function of the Interpereter.py file should be set to "Continuous"
4. From inside the "Source_Code" folder, execute "javac myinterpreter/*.java"
5. From inside the "Source_Code" folder, execute "java myinterpreter/Interpreter"
6. If single input is executed, the result will be shown in the terminal.
7. If continuous interactive input is executed, the terminal will show ">>>" which means the user can start giving expressions in the terminal to be executed. (To stop, enter "exit" or "stop")

3 Packages

The packages used are lang , Util and io which are the sub-packages of java package.

java.lang

The java.lang package contains the classes and interfaces that are fundamental to the core Java. Other classes from this package include: Boolean, Byte, Float, Math, Process and many more.

java.Util

This provides users with generic java utilities like collections framework, formatted printing and scanning, array manipulation utilities etc.

java.io

It is used for handling input and output operations. It contains classes for reading input from a stream of data and writing output to stream of data.

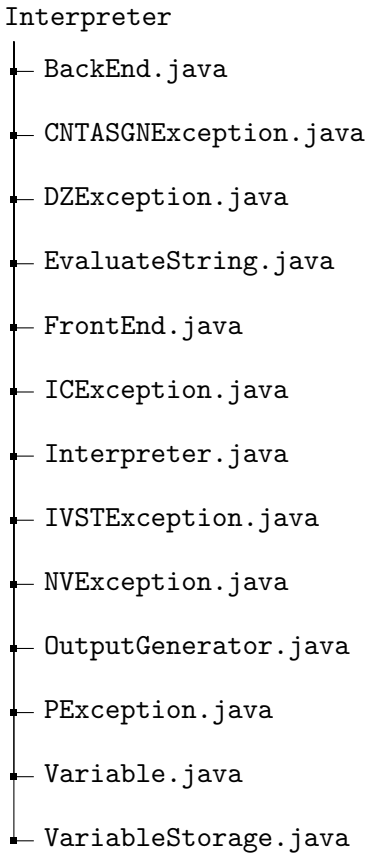
4 Exception Handling

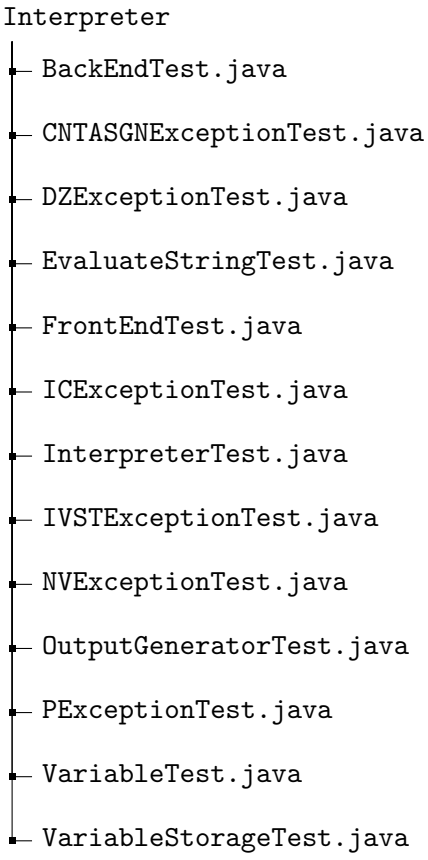
When evaluating expression, numerous exceptions are thrown based on the failure caused and in this project we tried to do implement few of the exceptions. Whenever the failure criteria for a respective exception is reached, we manually throw an exception and when we catch this exception, the corresponding error message is printed on the terminal.

The exceptions we are dealing with are:

1. "Cannot Assign to operator!"
2. "Divide by zero error!"
3. "Invalid character!"
4. "Invalid Syntax!"
5. "Variable does not exist!"
6. "Paranthesis not properly balanced!"

5 Design





5.a Interpreter

This is the class from where the program starts.

5.b FrontEnd

This class takes care of taking input from the user and passing the expression to be evaluated to the Backend class.

5.c BackEnd

This class validates the string and removes white spaces. The cleaned string is then given to the OutputGenerator class.

5.d OutputGenerator, EvalauteString

These 2 classes contains the logic for the expression evaluation.

5.e Variable

Variables given in expressions are expressed in form of this Variable class objects. This class contains methods to set/get variable names and their values.

5.f VariableStorage

This class contains logic for storing the different variables given in a single session of continuous interactive input from terminal.

5.g CNTASGNException, DZException, ICException, IVSTException, NVException, PException

These classes print the respective error messages in the terminal.