Name: Thanh Le
Class: CSC 413

Computer Science Department
San Francisco State University
CSC 413
Spring 2018

Assignment 1 - Expression Evaluator and Calculator GUI - **Documentation**

Link to github repository: https://github.com/csc413-02-sp18/csc413-p1-tle25

Project introduction and overview:

- For this project, we are able to run a basic calculator through EvaluatorUI or the main class.
- The task of this project is to create a basic calculator that can solve a simple math problems based on the arithmetic operations "#,+,-,*,/,^,!,()".
- In order to improve this calculator, the calculator has to know which operators will execute first. The logic is that each of the operators will have a specific priority. For example: the priority of each symbol is defined as "^" = 4; "*, /" = 3; "+;1" = 2; "!" = 1, and "(", ")" and "#" are set to 0.
- The idea is that when a mathematical expression is sent to Evaluator class, it will break and push each of tokens to Stacks. Assuming that we have a expression 1+2*3, the numbers 1,2, and 3 are pushed to operandStack, and "+" , "*" are pushed to operatorStack.
- Before the stack do the work, it has to check each of the tokens whether it is an operand or an operator. By doing that, I use the method tokenizer from Java library to break them apart.
- Operator and Operand classes have function named check. Its task is to check whether or not the token is valid.
- This project basically uses mathematical logic to solve the problem like a hierarchy from parentheses -> exponent -> multi/division -> add/sub.
- I also use Hashmap in this project by creating HashMah<Key, Value>. Key is a operator and Value is a executive function such as AdditionOperator, SubtractionOperator, etc.
- The program will keep evaluate the expression until operandStack has only one final value, then return it to where it is called.

Development Environment

- The program is developed in my laptop and written on Netbeans IDE 8.2
- We have a group discussion on slack where I can ask for help or someone when they get stuck.
- I also go to CS LAB to get help from others.

Instruction to compile and execute

First, download the all the scr files in Assignment #1

- **There is 3 options you can compile and execute with program:**

1/ You can run it under EvaluatorUI class which will display a panel with 20 buttons. With 0-9 digits and +,-,*,/,^,!, (,) Operators, “=” is to get answer, “CE” clear entry, and “C” button to erase all data.

Right click to Evaluator.java -> run file

2/ There are three testing classes with a specific expression each class.
SimpleExpressionTest.java, SimpleParenthese.java, HardTest.java
Right click to one of these three class you want to test, then run file.

3/ You also can compile and run this program with your own expression.
For example, if you want to compute an expression: 1+2*3; copy and paste it to String Val;

```
public class EvaluatorTester {
  public static void main(String[] args) {
                    Evaluator evaluator = new Evaluator();
                   String val = "1+2*3";
                   evaluator.eval(val);
                   System.out.println( val + " = " + evaluator.eval(val) );

        }
        }
```

Assumption

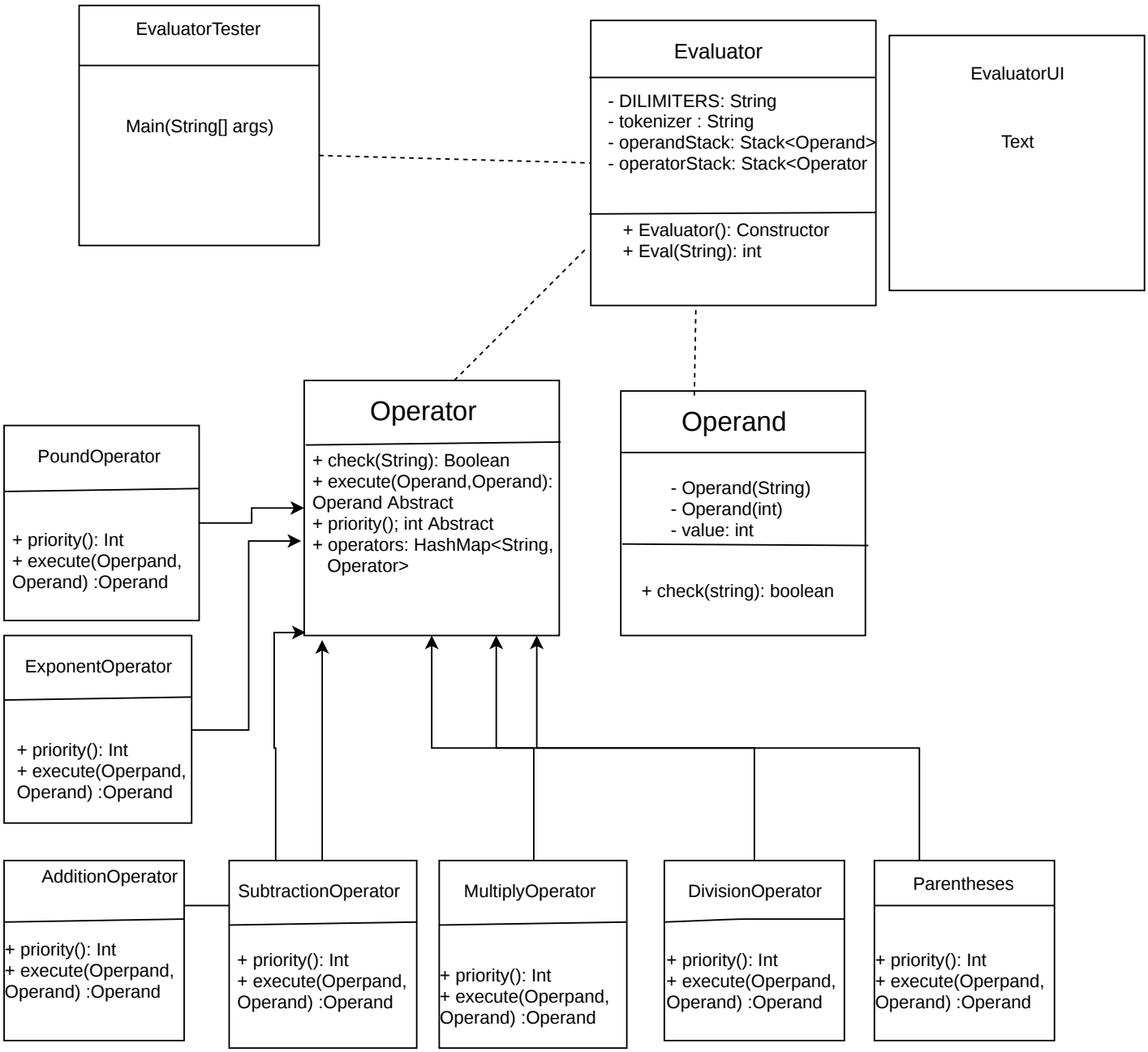
Most of the code are provided. I just need follow its skeleton to make it work.

Implementation discussion
-Diagram attached bellow

Results and conclusions

First, Welcome to Java world.

I am coming from C++, hence Java kinds of new to me. Most of the problems I had on this project was about systaxs most. About the logic, it mostly is similar with C++. I spent a lot of time on this Assignment, like reviewing Java language, watching Java tutorials from Youtube, finding and learning the meaning of each codes from java. Then I started to learn and understand the concept of each classes and methods from this assignment. At first, I was confused on the HashMap and abstract classes and methods because it is new to me. The hardest part challenging me was on the parentheses. It took me a lot of time to realize the problem. Finally, it works perfectly.


EvaluatorTester
Main(String[] args)
Evaluator
- DILIMITERS: String
- tokenizer : String
- operandStack: Stack<Operand>
- operatorStack: Stack<Operator
+ Evaluator(): Constructor
+ Eval(String): int
EvaluatorUI
Text
Operator
+ check(String): Boolean
+ execute(Operand,Operand): Operand Abstract
+ priority(); int Abstract
+ operators: HashMap<String, Operator>
Operand
- Operand(String)
- Operand(int)
- value: int
+ check(string): boolean
PoundOperator
+ priority(): Int
+ execute(Operpand, Operand) :Operand
ExponentOperator
+ priority(): Int
+ execute(Operpand, Operand) :Operand
AdditionOperator
+ priority(): Int
+ execute(Operpand, Operand) :Operand
SubtractionOperator
+ priority(): Int
+ execute(Operpand, Operand) :Operand
MultiplyOperator
+ priority(): Int
+ execute(Operpand, Operand) :Operand
DivisionOperator
+ priority(): Int
+ execute(Operpand, Operand) :Operand
Parentheses
+ priority(): Int
+ execute(Operpand, Operand) :Operand