**Tutorial No. 7**

**Problem statement:**

Implement interpreter design pattern for Calculator program.

**Design Assumptions:**

**Interpreter Design Pattern**

Intent

* Given a language, define a representation for its grammar along with an interpreter that uses the representation to interpret sentences in the language.
* Map a domain to a language, the language to a grammar, and the grammar to a hierarchical object-oriented design.

Problem

A class of problems occurs repeatedly in a well-defined and well-understood domain. If the domain were characterized with a "language", then problems could be easily solved with an interpretation "engine".

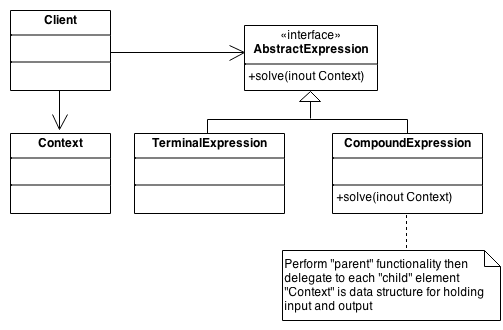
Discussion

The Interpreter pattern discusses: defining a domain language (i.e. problem characterization) as a simple language grammar, representing domain rules as language sentences, and interpreting these sentences to solve the problem. The pattern uses a class to represent each grammar rule. And since grammars are usually hierarchical in structure, an inheritance hierarchy of rule classes maps nicely.

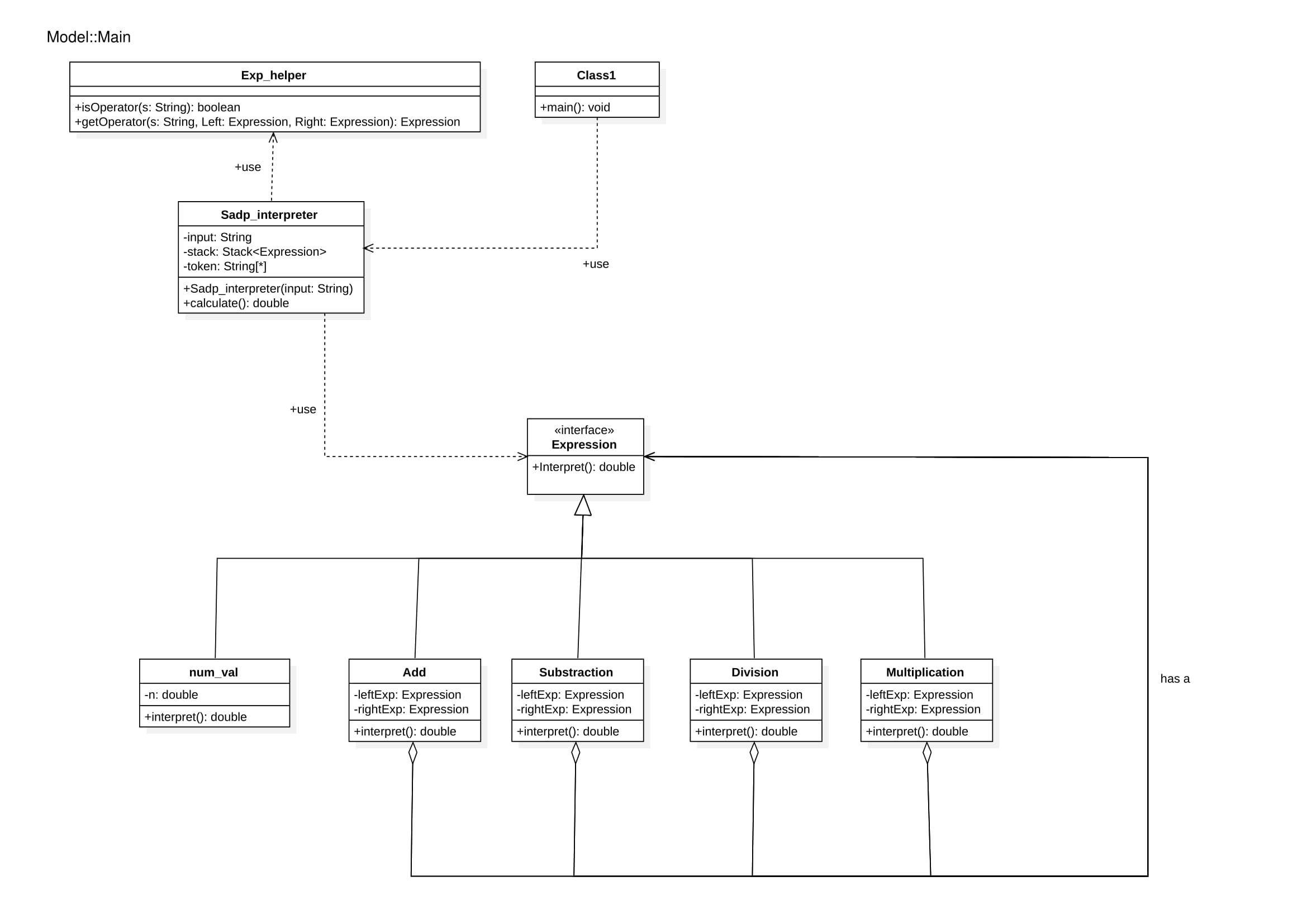
An abstract base class specifies the method interpret(). Each concrete subclass implements interpret() by accepting (as an argument) the current state of the language stream, and adding its contribution to the problem solving process.

Structure

Interpreter suggests modeling the domain with a recursive grammar. Each rule in the grammar is either a 'composite' (a rule that references other rules) or a terminal (a leaf node in a tree structure). Interpreter relies on the recursive traversal of the Composite pattern to interpret the 'sentences' it is asked to process.

****

**Design Diagrams:**

****

**Code:**

**Add.java**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package sadp\_interpreter;

/\*\*

\*

\* @author ROHIT NAVNATH JATHOT

\*/

public class Add implements Expression{

private final Expression leftExp;

private final Expression rightExp;

public Add(Expression leftExpression,Expression rightExpression ){

this.leftExp = leftExpression;

this.rightExp = rightExpression;

}

@Override

public double interpret() {

return leftExp.interpret() + rightExp.interpret();

}

}

**division.java**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package sadp\_interpreter;

/\*\*

\*

\* @author ROHIT NAVNATH JATHOT

\*/

public class division implements Expression{

private final Expression leftExp;

private final Expression rightExp;

public division(Expression leftExpression,Expression rightExpression ){

this.leftExp = leftExpression;

this.rightExp = rightExpression;

}

@Override

public double interpret() {

return leftExp.interpret() / rightExp.interpret();

}

}

**Exp\_helper.java**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package sadp\_interpreter;

/\*\*

\*

\* @author ROHIT NAVNATH JATHOT

\*/

public class Exp\_helper {

public static boolean isOperator(String s) {

if (s.equals("+") || s.equals("-") || s.equals("\*") || s.equals("/"))

return true;

else

return false;

}

public static Expression getOperator(String s, Expression left, Expression right) {

switch (s) {

case "+":

return new Add(left, right);

case "-":

return new Substract(left, right);

case "\*":

return new Product(left, right);

case "/":

return new division(left,right);

}

return null;

}

}

**Expression.java**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package sadp\_interpreter;

/\*\*

\*

\* @author ROHIT NAVNATH JATHOT

\*/

public interface Expression {

public double interpret();

}

c:\users\yoges\desktop\semester 6\sad alternate\tutorial 7\src\sadp\_interpreter\main.java

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package sadp\_interpreter;

/\*\*

\*

\* @author ROHIT NAVNATH JATHOT

\*/

public class main {

public static void main(String[] args) {

System.out.println(String.format("%.2f",new Sadp\_interpreter("77 3 /").calculate()));

}

}

**num\_val.java**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package sadp\_interpreter;

/\*\*

\*

\* @author ROHIT NAVNATH JATHOT

\*/

public class num\_val implements Expression{

private final double n;

public num\_val(double n){

this.n = n;

}

@Override

public double interpret() {

return n;

}

}

**Product.java**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package sadp\_interpreter;

/\*\*

\*

\* @author ROHIT NAVNATH JATHOT

\*/

public class Product implements Expression{

private final Expression leftExp;

private final Expression rightExp;

public Product(Expression leftExpression,Expression rightExpression ){

this.leftExp = leftExpression;

this.rightExp = rightExpression;

}

@Override

public double interpret() {

return leftExp.interpret() \* rightExp.interpret();

}

}

**Sadp\_interpreter.java**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package sadp\_interpreter;

import java.util.Stack;

/\*\*

\*

\* @author ROHIT NAVNATH JATHOT

\*/

public class Sadp\_interpreter {

/\*\*

\* @param args the command line arguments

\*/

private static String tokenString = "";

private static Stack<Expression> stack;

private static String[] tokenArray;

public Sadp\_interpreter(String tokenString){

this.tokenString = tokenString;

stack = new Stack<>();

tokenArray = tokenString.split(" ");

}

public double calculate() {

for (String s : tokenArray) {

if (Exp\_helper.isOperator(s)) {

Expression rightExpression = stack.pop();

Expression leftExpression = stack.pop();

Expression operator = Exp\_helper.getOperator(s, leftExpression,rightExpression);

double result = operator.interpret();

stack.push(new num\_val(result));

} else {

Expression i = new num\_val(Double.parseDouble(s));

stack.push(i);

}

}

return stack.pop().interpret();

}

}

**Substract.java**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package sadp\_interpreter;

/\*\*

\*

\* @author ROHIT NAVNATH JATHOT

\*/

public class Substract implements Expression{

private final Expression leftExp;

private final Expression rightExp;

public Substract(Expression leftExpression,Expression rightExpression ){

this.leftExp = leftExpression;

this.rightExp = rightExpression;

}

@Override

public double interpret() {

return leftExp.interpret() - rightExp.interpret();

}

}

**Output**

**Observation:**

Thus the interpreter design pattern was used in calculator program. We parsed the input string using interpreter pattern and produced the output.