```
/* Evaluating the expression "w x z - +" with w = 5, x = 10, and z = 42 *
    /

import java.util.HashMap;

public class InterpreterExample
{
    public static void main(String[] args)
    {
        String expression = "w x z - +";

        Evaluator sentence = new Evaluator(expression);
        HashMap<String,Expression> variables = new HashMap<String,
            Expression>();

        variables.put("w", new Number(5));
        variables.put("x", new Number(10));
        variables.put("z", new Number(42));

        int result = sentence.interpret(variables);
        System.out.println(result);
       }
}


import java.util.HashMap;
import java.util.Stack;

class Evaluator implements Expression
{
    private Expression syntaxTree;

    public Evaluator(String expression)
    {
        Stack<Expression> expressionStack = new Stack<Expression>();

        for (String token : expression.split(" "))
        {
            if(token.equals("+"))
            {
                Expression subExpression = new Plus(expressionStack.pop()
                    , expressionStack.pop());
                expressionStack.push( subExpression );
            }
            else if(token.equals("-"))
            {
                // it's necessary remove first the right operand from the
                    stack
                Expression right = expressionStack.pop();
                // ..and after the left one
                Expression left = expressionStack.pop();
                Expression subExpression = new Minus(left, right);
                expressionStack.push( subExpression );
            }
            else
                expressionStack.push( new Variable(token) );
        }
```

```
        syntaxTree = expressionStack.pop();
    }

    public int interpret(HashMap<String,Expression> context)
    {
        return syntaxTree.interpret(context);
    }
}


import java.util.HashMap;

interface Expression
    {
        public int interpret(HashMap<String,Expression> variables);
    }

class Number implements Expression
{
    private int number;

    public Number(int number)       { this.number = number; }

    public int interpret(HashMap<String,Expression> variables)  { return
        number; }
}

class Plus implements Expression
{
    Expression leftOperand;
    Expression rightOperand;

    public Plus(Expression left, Expression right)
    {
        leftOperand = left;
        rightOperand = right;
    }

    public int interpret(HashMap<String,Expression> variables)
    {
        return leftOperand.interpret(variables) + rightOperand.interpret
            (variables);
    }
}

class Minus implements Expression
{
    Expression leftOperand;
    Expression rightOperand;

    public Minus(Expression left, Expression right)
    {
        leftOperand = left;
        rightOperand = right;
    }
```

```
    public int interpret(HashMap<String,Expression> variables)
    {
        return leftOperand.interpret(variables) - rightOperand.interpret
            (variables);
    }
}

class Variable implements Expression
{
    private String name;
    public Variable(String name)       { this.name = name; }

    public int interpret(HashMap<String,Expression> variables)
    {
        if(null==variables.get(name)) return 0; //Either return new
            Number(0).
        return variables.get(name).interpret(variables);
    }
}
```