```
package com.interpreter;

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

public class Calculator {

  // 定义表达式
  private Expression expression;

  // 构造函数传参，并解析
  public Calculator(String expStr) { // expStr = a+b
    // 安排运算先后顺序
    Stack<Expression> stack = new Stack<>();
    // 表达式拆分成字符数组
    char[] charArray = expStr.toCharArray();// [a, +, b]

    Expression left = null;
    Expression right = null;
    //遍历我们的字符数组， 即遍历  [a, +, b]
    //针对不同的情况，做处理
    for (int i = 0; i < charArray.length; i++) {
      switch (charArray[i]) {
      case '+': //
        left = stack.pop();// 从stack取出left => "a"
        right = new VarExpression(String.valueOf(charArray[++i]));// 取出右表达式
"b"
        stack.push(new AddExpression(left, right));// 然后根据得到left 和 right 构建
AddExpresson加入stack
        break;
      case '-': //
        left = stack.pop();
        right = new VarExpression(String.valueOf(charArray[++i]));
        stack.push(new SubExpression(left, right));
        break;
      default:
        //如果是一个 Var 就创建要给 VarExpression 对象，并push到 stack
        stack.push(new VarExpression(String.valueOf(charArray[i])));
        break;
      }
    }
    //当遍历完整个 charArray 数组后，stack 就得到最后Expression
    this.expression = stack.pop();
  }

  public int run(HashMap<String, Integer> var) {
    //最后将表达式a+b和 var = {a=10,b=20}
    //然后传递给expression的interpreter进行解释执行
    return this.expression.interpreter(var);
  }
}
```