

| 学号             | 姓名  | 提交时间                | 总分  | 成绩 |
|----------------|-----|---------------------|-----|----|
| 20225102010114 | 丁俊杰 | 2023-11-02 16:34:33 | 100 | 60 |

1. 【简答题】完成一个简单的计算器功能：

得分：60/100



```

学生答案： String[] nums = {"1","9","2"};
            char[] chrs ={'+','X'};
            final String calculator = c.calculator(nums, chrs);
            System.out.println(calculator);
        }

/**
 * @:实现一个计算器,需要按照四则运算的法则来实现
 * 用户输入的是 操作数的数组 和运算符数组
 * 1 + 2 X 3 对应的就是 : 【1, 2, 3】 【'+', 'X'】
 * 2/7-3 对应的就是 : 【2,7,3】 【'/', '-'】
 */
public String calculator(String[] nums, char[] chrs) {
    if (nums.length == 0 || nums.length - chrs.length != 1) {
        return null;
    }
    if (nums.length == 1) {
        return nums[0];
    }
    for (int i = 0; i < chrs.length; i++) {
        if (chrs[i] == 'X' || chrs[i] == '/') {
            BigDecimal result;
            BigDecimal b1 = new BigDecimal(nums[i]);
            BigDecimal b2 = new BigDecimal(nums[i + 1]);
            if (chrs[i] == 'X') {
                result = b1.multiply(b2);
            } else {
                result = b1.divide(b2,4);
            }
            // 乘法（除法）位置在第一个
            if (i == 0) {
                if (nums.length == 2) {
                    return result.toString();
                } else {
                    String[] newNum = new String[nums.length - 1];
                    newNum[0] = result.toString();
                    for (int j = 1; j < newNum.length; j++) {
                        newNum[j] = nums[j + 1];
                    }
                    return calculator(newNum, new String(chrs).substring(1).toCharArray());
                }
            } else {
                // 乘法（除法）位置非第一个
                String[] newNum = new String[nums.length - 1];
                for (int j = 0; j < i; j++) {
                    newNum[j] = nums[j];
                }
                newNum[i] = result.toString();
                for (int j = i + 1; j < newNum.length; j++) {
                    newNum[j] = nums[j + 1];
                }
                final char[] chars = (new String(chrs).substring(0, i) + new String(chrs).substring(i + 1)).toCharArray();
                return calculator(newNum, chars);
            }
        }
    }
    BigDecimal result = new BigDecimal(nums[0]);
    for (int i = 0; i < chrs.length; i++) {
        if (chrs[i] == '+') {
            final BigDecimal bigDecimal = new BigDecimal(nums[i + 1]);
            result = result.add(bigDecimal);
        }
        if (chrs[i] == '-') {
            final BigDecimal bigDecimal = new BigDecimal(nums[i + 1]);
            result = result.subtract(bigDecimal);
        }
    }
    return result.toString();
}

```

参考答案：完成一个简单的计算器功能