

# 软件测试课程第一次作业要求

2019. 9

采用边界值测试以及等价类测试方法（均需支持单缺陷假设）对以下软件需求进行测试用例设计，需求说明如下：

## 1. 软件目标

通过设计一个简单的**数学表达式计算器**，要求设计者以面向对象程序的设计方式来实现计算器的具体功能。

## 2. 软件要求

### 2.1 业务背景

#### (1) 表达式定义

本程序表达式设定为可由多个运算符和数字组成，可以含括号，为了降低题目难度，括号只用于标注数字为正数还是负数，同时，假设操作数是整数，运算符只有四种类型：+、-、\*和/，例如如下表达式： $-2+3*(-5)/3-2*9$ 。

#### (2) 有效表达式校验

在对表达式求值运算之前，必须先对表达式的正确性进行校验。例如：空字符串、运算符连续、空括号、括号不配对、右括号前是运算符、“（”不为第一个字符且前面不是运算符、“）”不为最后一个字符且后面不是运算符、表达式内含非法字符等。（**注意，编程者需根据实际情况厘清表达式字符串规则，从而设定具体的校验准则，以上只是部分提示**）

### 2.2 运算规则

为简单起见，表达式计算不考虑运算符的优先级，如果表达式正确，则从左到右依次计算，最终得出结果，其中，除法时，遵从 C 语言中整数除法的运算规则（即取整），另外，一旦出现除零操作，程序要给出相应错误提示。

## 3. 作业内容和成果物

### 3.1 作业内容

实现一个符合 2.1 节和 2.2 节所描述的表达式计算器 java 程序。

### 3.2 提交内容

#### (1) Java 语言程序（整个 Eclipse 工程文件夹），且该工程必须已经通过编译且能够运行；

(2) 程序说明文档 (Readme, word 文件, 相当于作业报告, 放置在 Eclipse 工程文件夹中即可),  
内容包括:

- 1) 表达式计算器运算规则和程序功能说明;
- 2) 程序的输入说明, 包括标准输入格式、输入限制和遇见输入错误时的响应信息;
- 3) 程序计算结果的输出规格, 以及可预见的运行错误响应信息;

## 4. 软件要求和限制

### 4.1 输入规范

**本次工程项目要求必须在控制台进行输入。**

用户输入**欲要计算的表达式的字符串**;

所有的符号应采用 ASCII 字符集中的英文符号, 而不是中文字符。例如括号应该是“()”, 而不是中文的“( )”。**每次只能输入一个请求表达式**。一条请求的内部元素之间 (包括表达式头、尾) 可以有空格, 要求程序能够自动过滤。

本次项目要求一次性输入完整请求 (回车即为输入完毕), 然后执行程序进行表达式计算并输出结果。假如**标准输入的请求是不符合表达式规范的, 则直接按照 4.2 的输出规范进行输出**。

### 4.2 输出规范

程序的输出为**表达式求值结果的描述**, 且必须在控制台进行输出。

格式为 (示例) : **-2+3\*(-5)/3-2\*9=-27**

**注意: 输出结果只能写在一行上, 假如一个结果写成多行 (有回车), 则测试脚本无法识别!**

**输出格式要求采用 UTF-8 标准。**

**对于本程序不能识别的输入内容, 必须按照以下规格进行输出:**

**FORMAT ERROR!**

**对于运算过程中出现除 0 情况, 必须按照以下规则进行输出:**

**Error: Divided by zero!**

相关的输出规范要在**说明文档 (README)** 中加以说明。包括每种 Format Error 的情况。

其他未规定的地方可由编程者自行决定。但是需要在 readme 中说明。

### 4.3 设计要求

本次工程的表达式求值方法务必命名为 calculate, 其方法原型如下:

```
//expression 为输入的待计算表达式;  
//return String 为输出结果。
```

```
public String calculate(String expression);
```

否则测试程序无法识别。

本次工程将使用的相关知识：

- (1) 字符串操作。
- (2) 使用[正则表达式](#)对表达式进行合法性校验。
- (3) Java 程序的三种控制结构（顺序、选择、循环）。

## 5. 其它说明事项

### 5.1 设计建议

- (1) 从控制台输入表达式运算；
- (2) 对读取的字符串进行合法性校验；
- (3) 解析合法字符串内容；
- (4) 进行表达式计算；
- (5) 对结果进行控制台输出。

### 5.2 Tips

如何设计程序结构，使得每个类和方法的职责相对单一且简单：表达式、表达式校验、表达式计算等，它们的定义是什么，有什么（属性）；它们能做什么，在什么情况下做（方法）。假如程序进行继续迭代，当前设计该如何尽可能适应接下来的迭代。

## 6. 其他规定

- (1) 文档中**粗体字体部分**为强制要求。
- (2) 无效工程，以下四种情况视为无效工程。
  - 1) 程序不能编译和运行；
  - 2) 未使用 Java 语言；
  - 3) 所编制的程序不是本工程的内容。
  - 4) 无法通过任何一个可以输出正常结果的公共测试案例；
  - 5) 测试程序无法识别。

以上即为完整的软件功能需求，请大家使用 Excel 进行测试用例的设计，基本格式如下：

测试用例设计分析：（测试用例设计的分析过程，边界值和等价类可以分开写）

ID	输入	预期输出	实际输出	测试结果	备注			
1					例如所覆盖的等价类情况			
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								

作业提交要求：

每名同学交一份 Excel 文件，文件命名为“学号-姓名-第一次作业.xlsx”，发送到超星上该门课程，具体见超星要求：<http://nchu.fanya.chaoxing.com/portal>