# 设计文档

UI使用Java编写，算法使用C语言编写。两者通过JNI进行交互。

## UI

主界面参考谷歌原生计算器,实现如下:

状态栏:用户输入的时候显示:正在计算,输入完成计算,显示为空.

编辑输入栏,用于用户输入矫正表达式,其中可以智能判断输入的表达式,如果出错,给用户错误提醒.

结果输出栏,数字计算,直接显示,包含字母的提示到结果界面显示.

键盘输入:输入包含:0-9,abcdlmnxyz π(键盘中用PI表示)和e(键盘中用EXP表示)的输入.其中字母输入使用黄色高亮显示,且键盘实现滚动显示

符号输入:算术符:(单击)"÷", "×", "-", "+", (长按)"!" "√", "^", "()", "[", "]" 算术符输入实现蓝色高亮显示,实现滚动显示.

DEL按钮,点击实现删除一个字符,长按实现清屏功能.

菜单栏实现两个功能,点击可以查看应用信息,可以退出程序.

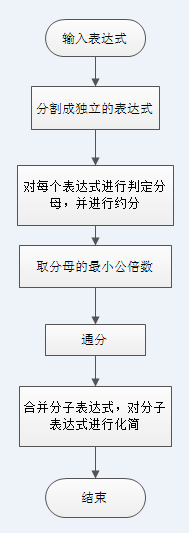
## 算法

### 数字计算器

通过出栈、入栈操作将中缀表达式转换为后缀表达式进行计算。设计思路比较常见，故此省略。

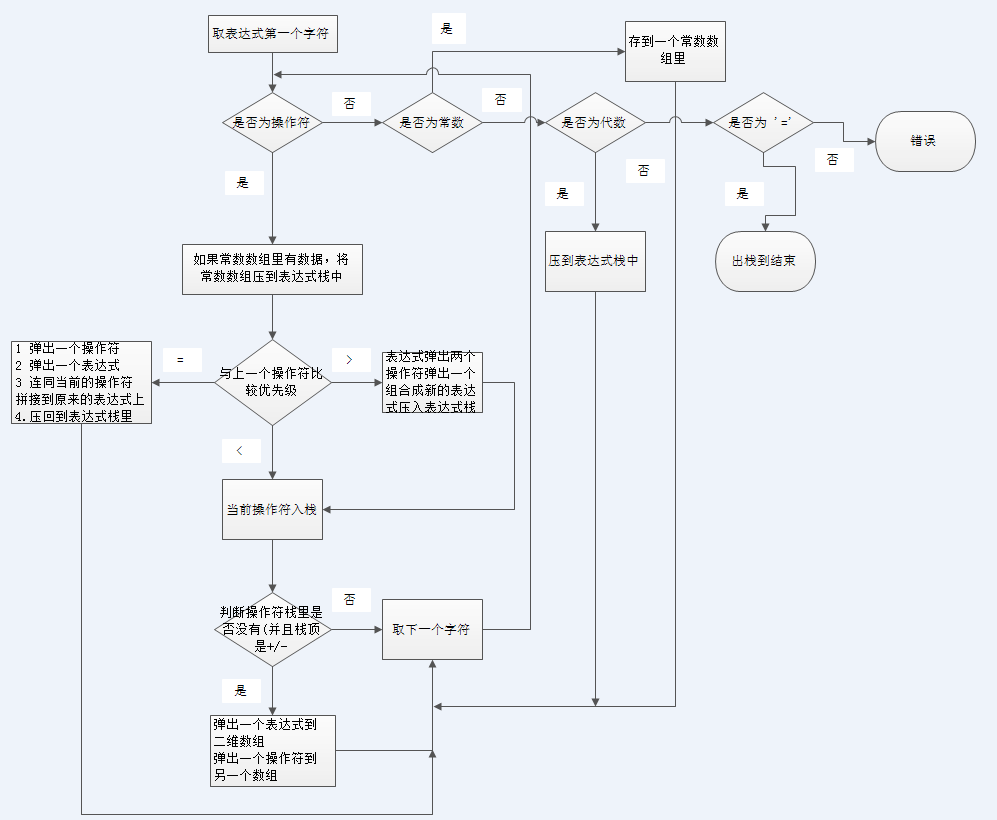
### 代数计算器

#### 2.1 总体设计思路

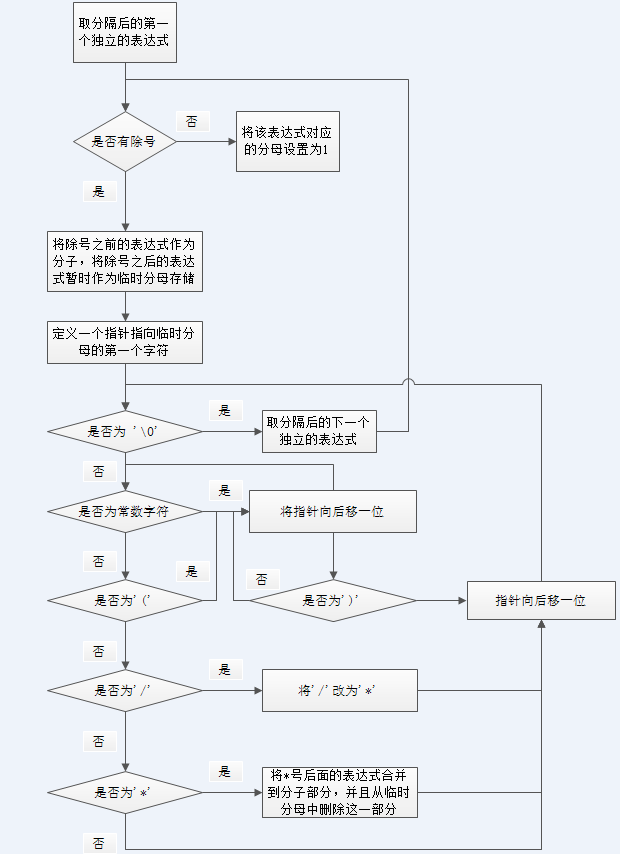


#### 2.2 细节设计思路

2.2.1 分隔成独立的表达式



2.2.2 对每个表达式进行判定分母（由于时间关系，约分尚未完成）



2.2.3取分母的最小公倍数

a. 使用一个数组保存取分隔后的第一个独立的表达式的分母部分

b. 将这个数组与第二个独立的表达式的分母部分进行比较

c. 相同的部分忽略，不相同的部分追加到数组中

d. 依次将这个数组与所有的独立的表达式的分母部分比较，最终这个数组就是所需的最小公倍数

2.2.4通分

a. 将上一步得到的最小公倍数与每个独立表达式的分母部分比较

b. 相同的部分忽略，不同的部分追加到该表达式的分子部分

c. 最终所有的分母部分都一样，下一步将分子部分化简

2.2.5合并分子表达式，对分子表达式进行化简

未完成