# 需求分析

编写一个计算器工具，能够实现windows附件中科学型计算器的功能。

# 概要和详细设计

程序功能：

本程序实现了一个计算器，

支持运算+-\*/%,tan,arctan(atan),cos,arccos(acos),sin,arcsin(asin),log,还有^(指数)，

使用double实现，并且支持括号，此外还支持圆周率（输入时用p字符表示）和自然底数e。

设计概要：

首先我们都知道可以用栈的数据结构来解决。

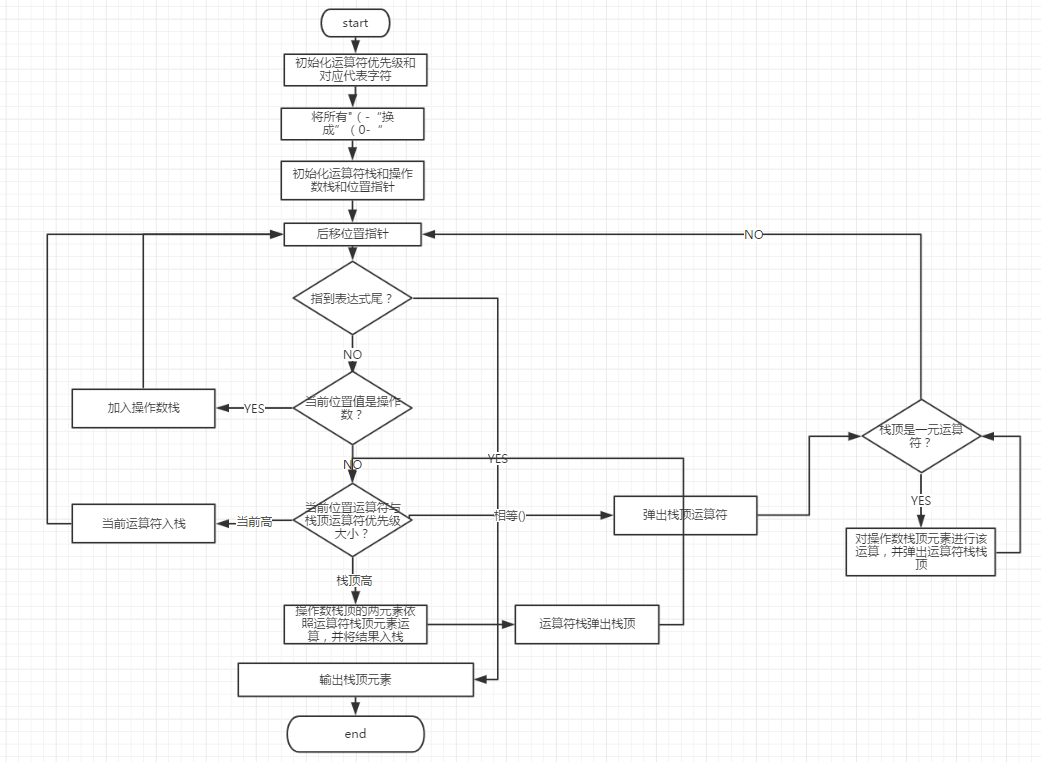
做法是设置两个栈，运算符栈和操作数栈，再给每个运算符定义两个（栈内栈外）优先级。

从前向后扫表达式，数字或常量（p\e）直接加入操作数栈，而运算符的话要同栈顶操作数比较优先级，栈顶低就入栈；相等就是()的清形，“(”出栈即可；栈顶高就可以将操作数栈的前两元素弹出进行运算，再将运算结果入栈，然后继续比较当前算符和栈顶算符的优先级，并根据大小重复上述工作。

实际实现时有些注意点：

1. 我们输入的表达式可能有-1-2或1\*(-1)这样的，即-号变成了一个一元运算符，我的做法是预处理表达式把这样的情况转化为0-1-2或1\*(0-1)，将-变回二元表达式。
2. 由于tan\atan\log\sin\asin\cos\acos的出现，一元运算变成了必然，我是特判两个运算符优先级相等（即左右括号）的情况，除了将左括号出栈，还要检查前面是否是一元运算符，若是则将该算符弹出，将操作数栈栈顶元素取出进行该运算，直到算符栈栈顶算不是一元算符为止
3. 我还加入了适当的除0检查，避免程序因除0而崩溃；对于log也有底数大于0的检查。

# 程序结构



# 代码解释

Line 13 to 16:定义函数dcmp(double x)当x的绝对值与0的差距小等于eps时返回0

Line 18 to 38:定义并设置(setLevel)每个运算符在栈内(levelIn)和栈外(levelOut)的优先级。

Line 42 to 49:将输入表达式中会出现的长字符串对应的处理时使用的单字符映射建好。

Line 50 to 64:solveIn()预处理表达式将所有(-的情形（当然表达式以-开头也要这样处理）变成(0-.

Line 82 to 83:p字符表示π，e字符表示自然底数

Line 84 to 92:处理是数字的情况。

Line 95 to 101:处理tan,atan,cos.log等长度较长的运算符转换为对应的单字符。

Line 106 to 125:栈顶和当前字符的优先级相等（左右括号）除了丢弃这个外，还需检查前面的运算符是否是单运算符，若是要继续运算操作数栈栈顶元素。

# 测试

注释掉的是输出

-cos(p/2)-sin(p/2)

//-1.00000

-log(e)

//-1.00000

(cos(324834))^2+(sin(324834))^2

//1.00000

3\*10^3

//3000.00000

3\*10%3

//0.00000