## 产品概述

**计算器：**

它是一个拥有扁平化优雅用户界面的科学计算器，拥有科学计算与基础计算器功能，可以计算是是数学表达式：从一个简单的表达式，如“1+2”到更高级的表达式，如“3+（3\*9（80+sin90））” 除了可以实现了系统自带计算器功能外还可以进行基本函数的运算和括号的运算，能根据表达式的优先级来自动区分运算顺序。

它可以记住你的计算，添加强大的容错处理功能，如点击空小数点默认为0，除数不能为0，log、根号与sin等符号中的数不能为负等。

## 面向人群

高中生,大学生等

## 产品版本及兼容

适用于ANDROID 操作系统

目标API 4.1

向下兼容 API 4.0

## 思想

**计算器：**

**计算表达式过程：**

* 获取用户点击按钮输入的表达式的字符串，检验合法性，通过

StringTokenizer expToken = **new** StringTokenizer(expression,

"+-×÷()sctgl!√^");

截取表达式

* 从左向右扫描，数字入number栈，运算符入operator栈
* 优先级定义： +-基本优先级为1，×÷基本优先级为2，log ln sin cos tan n!基本优先级为3，√^基本优先级为4 ，括号内层运算符比外层同级运算符优先级高4
* 当前运算符优先级高于栈顶压栈，
* 低于栈顶弹出一个运算符与两个数进行运算
* 重复直到当前运算符大于栈顶
* 扫描完后对剩下的运算符与数字依次计算

## 设计

**计算器：**

**用户界面布局：**

1. **输入模块，**主要描述了计算器键盘以及键盘上的监听即主要负责读取用户的键盘输入以及相应触屏的按键，当用户点击按键或者屏幕的时候监听器会去调用相应的处理办法，具体的处理办法会在相关的地方进行说明。在本项目中，在触屏上提供了各种功能的按键，分别是C,0,1,2,3,4,5,6,7,8,9，+，-，\*，/，=，MC，根号，平方，退格，退出
2. **显示模块，**主要描述了计算器的显示区，即该区域用于显示用户输入的数据以及最终的计算结果和一些其他信息。

**功能实现：**

1. **计算模块的设计的实现**，计算模块主要描述了计算器的整体，实现了计算器的界面，定义了用户单击的处理办法，通过定义的处理办法来完成用户的操作命令，实现用户的计算要求，这些要求包括了数值的输入，小数点的输入，正负号的输入，记忆相关功能的完成，加，减，乘，除等各计算功能的完成，运算符的替换，初始化系统等等。

主要类只有一个，里面还有一个内部类：

**主要类即：**

public class MainActivity extends Activity

**该类包含的属性：**

private Button[] btn// 0~9十个数字

private EditText input;// 用于显示输出结果

private TextView mem, \_drg;//mem显示memory中所存的表达式及结果，\_drg:当前是角度还是弧度（DRG是角度，RAD是弧度）

private Button div, mul, sub, add, equal, sin, cos, tan, log, ln, sqrt,square, factorial（阶乘）, bksp（退格）, left, right, dot, exit, drg（角度弧度键）, mc, c;

public String str\_old;//原来输入的字符串

public String str\_new;//为了便于计算而简化后的字符串

public boolean vbegin = true;// 控制输入，true为重新输入，false为接着输入（清屏后输入

和继续输入）

public boolean drg\_flag = true;// true为角度，false为弧度

public double pi = 4 \* Math.atan(1);// π值

public boolean tip\_lock = true;// true为正确，可以继续输入，false错误，输入锁定

public boolean equals\_flag = true;// 是否在按下=之后输入，true为之前，false为之后

**方法：**

protected void onCreate(Bundle savedInstanceState)

//运行时调用

InitWigdet();

//初始化所有的组件

private void AllWigdetListener();//为所有按钮绑定监听器

//命令捕捉

private OnClickListener actionPerformed = new OnClickListener()

新建一个对象

public void onClick(View v) {

* 按键上的命令获取
* 显示器上的字符串
* 检测输入是否合法
* 检测显示器上的字符串是否合法（调用后面的right（）函数），若输入正确，则将输入信息显示到显示器上

1. 如果点击了“DRG”，则切换当前弧度角度制，并将切换后的结果显示到按键上方。
2. 如果输入时退格键，并且是在按=之前： 一次删除3个字符，2个，1个，之前输入不合法则删除全部（调用后面的right（）函数判断是否合法，调用后面的函数private int TTO (String str)判断是一个两个还是三个）

3.如果是在按=之后输入退格键则将显示器内容设置为0

4.如果输入的是清除键则将显示器内容设置为0，重新输入标志置为true，表明可以继续输入，表明输入=之前

5.如果输入的是”MC“，则将存储器内容清0

6.如果按”exit“则退出程序：弹出一个对话框“请确认记录必要数据后再退出系统”

7.如果输入的是=号，并且输入合法（调用后面的right（）函数）表明不可以继续输入，表明输入=之后，保存原来算式样子，替换算式中的运算符，便于计算，重新输入标志设置true

* 将-1x转换成-
* 计算算式结果（调用calc（）里面的函数），表明可以继续输入

private void print(String str) //将信息显示在显示屏上

**核心：**

**内部类public class calc** {

**1**. **public calc()//构造函数**

final int MAXLEN = 500;//栈的大小

**2.**   **public void process(String str)**

**计算过程：**

* 获取用户点击按钮输入的表达式的字符串，检验合法性，通过

StringTokenizer expToken = **new** StringTokenizer(expression,

"+-×÷()sctgl!√^");

截取表达式

* 从左向右扫描，数字入number栈，运算符入operator栈
* 优先级定义： +-基本优先级为1，×÷基本优先级为2，log ln sin cos tan n!基本优先级为3，√^基本优先级为4 ，括号内层运算符比外层同级运算符优先级高4
* 当前运算符优先级高于栈顶压栈，
* 低于栈顶弹出一个运算符与两个数进行运算
* 重复直到当前运算符大于栈顶
* 扫描完后对剩下的运算符与数字依次计算

**·变量**：

int weightPlus = 0, topOp = 0, topNum = 0, flag = 1, weightTemp = 0;// weightPlus为同一（）下的基本优先级，weightTemp临时记录优先级的变化

// topOp为weight[]，operator[]的计数器；topNum为number[]的计数器

// flag为正负数的计数器，1为正数，-1为负数

int weight[]; // 保存operator栈中运算符的优先级，以topOp计数

double number[]; // 保存数字，以topNum计数

char ch, ch\_gai, operator[];// operator[]保存运算符，以topOp

计数

String num;// 记录数字，str以+-×÷()sctgl!√^分段，+-×÷()sctgl!√^字符之间的字符串即为数字

weight = new int[MAXLEN];//符号栈中符号对应的优先级

number = new double[MAXLEN];

operator = new char[MAXLEN];

String expression = str;

StringTokenizer expToken = new StringTokenizer(expression,"+-×÷()sctgl!√^");//按符号分割字符串

**取得字符串后:**

1. 判断正负数
2. 取得数字，并将正负符号转移给数字
3. 取得整个数字
4. 将指针退回之前的位置
5. 将正负符号转移给数字
6. 计算运算符的优先级
7. 如果当前优先级大于堆栈顶部元素，则直接入栈
8. 否则将堆栈中运算符逐个取出，直到当前堆栈顶部运算符的优先级小于当前运算符（取出数字数组的相应元素进行运算（ sin等三角函数计算时要进行角度弧度的判断及转换））

I.依次取出堆栈的运算符进行运算

**得到结果后：**

1. 如果是数字太大，提示错误信息
2. 输出最终结果
3. public double FP(double n) ;

FP = floating point 控制小数位数，达到精度 否则会出现

0.6-0.2=0.39999999999999997的情况，用FP即可解决，使得数为0.4 本格式精度为15位

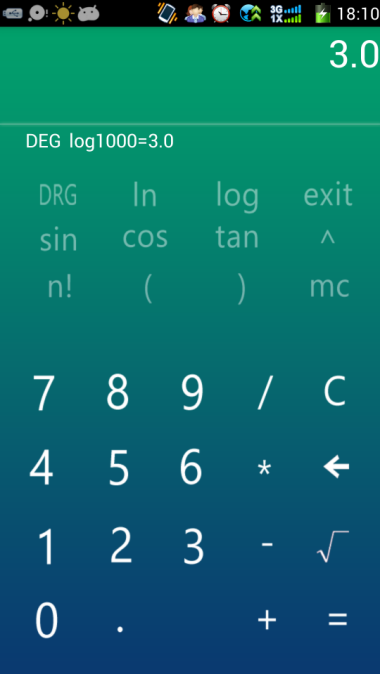
**4.**public double N(double n);//阶乘算法

**5.**public void showError(int code, String str)//错误提示，按了"="之后，若计算式在process()过程中，出现错误，则进行提示

## 调试及测试

**计算器：**

**各类表达式计算与清零：**



**容错处理与退出：**



## 不足与改进方向

1. 允许输入以0开头的数字，如“00000006+00002=8”，我们因该该禁止无效数字0的输入
2. 允许输入“0.0000000000000000000000000000000000000000000000000000001”从而超过了显示的范围造成数字不能完全显示，应该规定最长输入数，如果超出就警告；
3. 不能进行科学计数法的输入，就是

如果想计算“0.0000000000000000000000000000000000000000000000000000001+2”

因为前面一个数位数太多，会输入不进去，所以就不能计算，应该提供科学计数法的

输入，如0.1\*E-30+2 等。

1. 不能计算 “0除以0”，按照 matlab 标准，应该为NaN ,即无解，我们显示为“0不能做除数”
2. 如果连续输入多个符号，应该不断变化显示，并且最终显示最后一个符号，并且按照最后一个符号来计算，如输入“8/\*-+++++-+\*\*\*2”，显示的地方不断变化，最终显示“8\*2”，

并且计算结果为16，其实这样就方便了用户使用，例如本来想输入“8\*2”，结果输入了“8/”，符号输入错误了，这时候一般还要 输入退格键，这样很麻烦，如果可以再输入一个“\*”，并且覆盖错误输入的“/”，就方便多了。

1. 主要按钮应该大一点，，并且占据主要位置，如等于号，加号
2. 计算器界面风格为扁平化，背景炫酷但是按钮的位置不清晰，造成年龄较大的用户使用不方便，应该把主要按钮的轮廓勾勒出来，并且增加点击后效果。