计算器的设计与实现说明文档

完成人：张帆　学号：20192131077

完成时间：2021 月 11 月 5 号

一、软件名称

安卓计算器

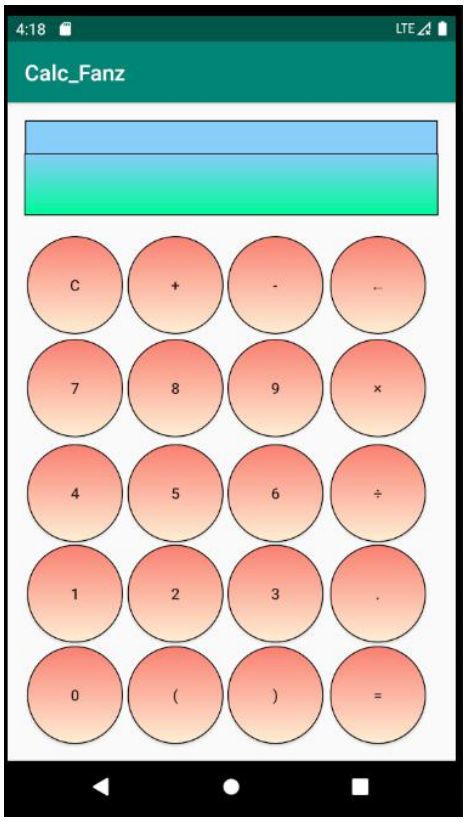
二、软件内容简介

这是一个使用 Android Stduio 编写的安卓计算器，能够实现的功能有加、减、乘、除以及括号的简单四则运算。软件的空间使用了自己编写的三种 drawable 文件，整体看上去软件的界面风格是较为简约清爽的。

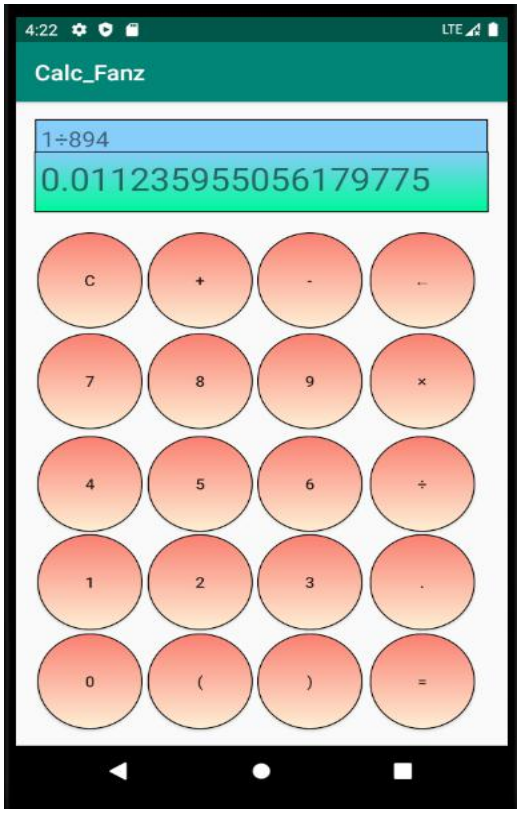
软件的 Ui 设计模块是由 Kotlin 语言编写的，而具体的计算模块是由 Java 语言编写的。先读取屏幕上由用户输入的计算式,再把这个计算式转化为后缀表达式,最后计算得出结果,再输出到屏幕上。

三、界面设计

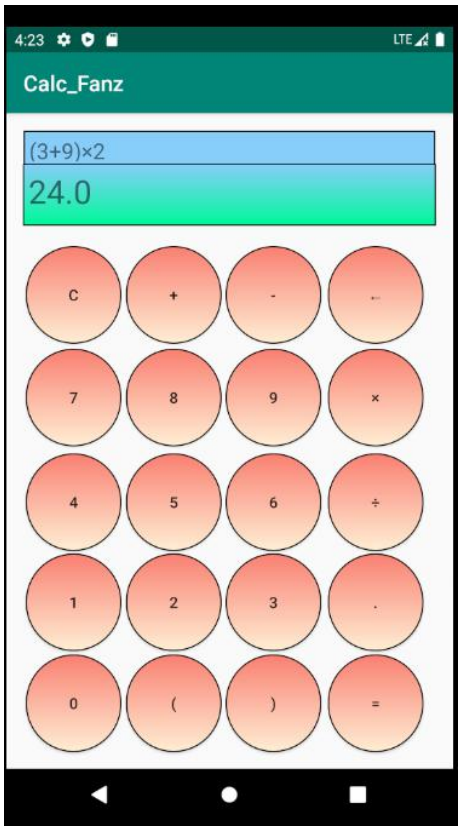
3.1 主界面



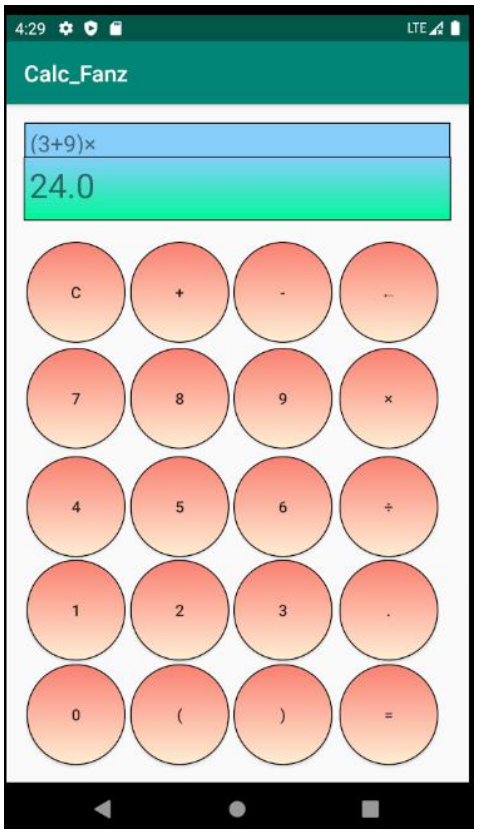
3.2 小数计算



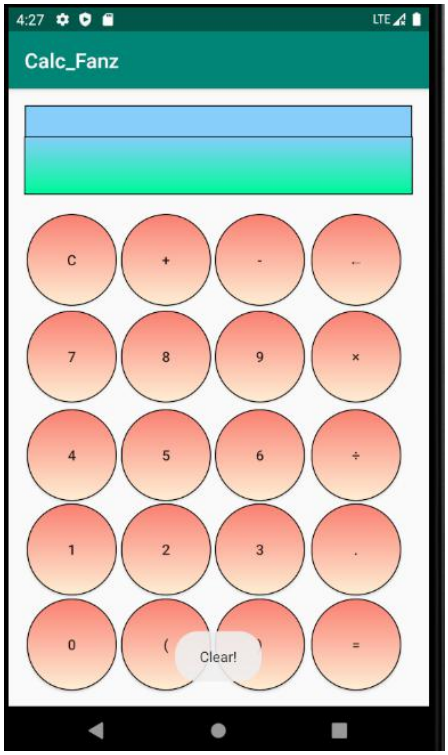
3.3 带括号的计算



3.4 后退一位



3.5 清空



四、代码设计

4.1 MainAcitivy.kt

```
package com.example.calc_fanz

import com.example.calc_fanz.Calculator
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_main.*

import java.math.BigDecimal
import java.util.ArrayList
import java.util.Stack
import androidx.core.app.ComponentActivity
import androidx.core.app.ComponentActivity.ExtraData
import android.icu.lang.UCharacter.GraphemeClusterBreak.T
import android.util.Log
import android.widget.Toast
import androidx.core.content.ContextCompat.getSystemService
import java.sql.SQLOutput
```

```
class MainActivity : AppCompatActivity()
{
    override fun onCreate(savedInstanceState: Bundle?)
    {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        button1.setOnClickListener {
            textView.setText("")
            textView2.setText("")
            Toast.makeText(this, "Clear!", Toast.LENGTH_SHORT).show()
        }
        button2.setOnClickListener {
            val currentnumber = textView.getText().toString()
            val newnumber = currentnumber + "+"
            textView.setText(newnumber)
        }
        button3.setOnClickListener {
            val currentnumber = textView.getText().toString()
            val newnumber = currentnumber + "-"
            textView.setText(newnumber)
        }
        button4.setOnClickListener {
            var newnumber = ""
            val currentnumber = textView.getText().toString()
            if (currentnumber.length != 0){
                for (i in 0..currentnumber.length - 2){
                    newnumber += currentnumber[i]
                }
                textView.setText(newnumber)
            }
        }
        button5.setOnClickListener {
            val currentnumber = textView.getText().toString()
            val newnumber = currentnumber + "7"
            textView.setText(newnumber)
        }
        button6.setOnClickListener {
            val currentnumber = textView.getText().toString()
            val newnumber = currentnumber + "8"
            textView.setText(newnumber)
        }
        button7.setOnClickListener {
```

```
        val currentnumber = textView.getText().toString()
        val newnumber = currentnumber + "9"
        textView.setText(newnumber)
    }
    button8.setOnClickListener {
        val currentnumber = textView.getText().toString()
        val newnumber = currentnumber + "×"
        textView.setText(newnumber)
    }
    button9.setOnClickListener {
        val currentnumber = textView.getText().toString()
        val newnumber = currentnumber + "4"
        textView.setText(newnumber)
    }
    button10.setOnClickListener {
        val currentnumber = textView.getText().toString()
        val newnumber = currentnumber + "5"
        textView.setText(newnumber)
    }
    button11.setOnClickListener {
        val currentnumber = textView.getText().toString()
        val newnumber = currentnumber + "6"
        textView.setText(newnumber)
    }
    button12.setOnClickListener {
        val currentnumber = textView.getText().toString()
        val newnumber = currentnumber + "÷"
        textView.setText(newnumber)
    }
    button13.setOnClickListener {
        val currentnumber = textView.getText().toString()
        val newnumber = currentnumber + "1"
        textView.setText(newnumber)
    }
    button14.setOnClickListener {
        val currentnumber = textView.getText().toString()
        val newnumber = currentnumber + "2"
        textView.setText(newnumber)
    }
    button15.setOnClickListener {
        val currentnumber = textView.getText().toString()
        val newnumber = currentnumber + "3"
        textView.setText(newnumber)
    }
```

```
        button16.setOnClickListener {
            val currentnumber = textView.getText().toString()
            val newnumber = currentnumber + "."
            textView.setText(newnumber)
        }
        button17.setOnClickListener {
            var tempnumber = ""
            var currentnumber = textView.getText().toString()

            for (i in 0..currentnumber.length - 1)
            {
                if (currentnumber[i] == '×')
                {
                    tempnumber += "*"
                }
                else if (currentnumber[i] == '÷')
                {
                    tempnumber += "/"
                }
                else
                {
                    tempnumber += currentnumber[i]
                }
            }
            var newnumber = Calculator.resu(tempnumber)
            textView2.setText(newnumber)
        }
        button18.setOnClickListener {
            val currentnumber = textView.getText().toString()
            val newnumber = currentnumber + "0"
            textView.setText(newnumber)
        }
        button19.setOnClickListener {
            val currentnumber = textView.getText().toString()
            val newnumber = currentnumber + "("
            textView.setText(newnumber)
        }
        button20.setOnClickListener {
            val currentnumber = textView.getText().toString()
            val newnumber = currentnumber + ")"
            textView.setText(newnumber)
        }
    }
}
```

4.2 Calculator.java

```
package com.example.calc_fanz;

import java.math.BigDecimal;
import java.util.ArrayList;
import java.util.Stack;

public class Calculator
{
    public static int priority(char a)
    {//判断符号优先级
        switch (a) {
            case '+':
            case '-':
            case '(':
                return 1;
            case '*':
            case '/':
                return 2;
        }
        return 0;
    }

    public static char[] middleToBack(String exp)
    {//中缀表达式转后缀表达式
        Stack<Character> stack = new Stack();//栈用来进出运算的符号
        char[] arr = exp.toCharArray();
        char[] brr = new char[arr.length * 2];//保存后缀表达式，需在数字和符号之间加空格区分，
因此长度定为arr的2倍
        int count = 0;//标记brr

        for (int i = 0; i < arr.length; i++)
        {
            if (i == arr.length - 1) {//防止越界
                brr[count++] = arr[i];
                brr[count++] = ' ';
            } else if ((arr[i] >= '0' && arr[i] <= '9') && !(arr[i + 1] >= '0' && arr[i + 1] <=
'9')) {
                brr[count++] = arr[i];
                brr[count++] = ' ';
            } else if (arr[i] >= '0' && arr[i] <= '9') {
                brr[count++] = arr[i];
            } else {//符号
                if (stack.isEmpty()) {
```

```
                    stack.push(arr[i]);
                } else if (arr[i] == '(') {
                    stack.push(arr[i]);
                } else if (arr[i] == ')') {
                    while (stack.peek() != '(') {//'('与')'之间的符号出栈
                        brr[count++] = stack.peek();
                        brr[count++] = ' ';
                        stack.pop();
                    }
                    stack.pop();//将'('出栈
                } else if (priority(arr[i]) >= priority(stack.peek())) {//如果优先级高于或等于栈顶元素，直接入栈
                    stack.push(arr[i]);
                } else if (priority(arr[i]) < priority(stack.peek())) {//如果优先级低于栈顶元素，依次出栈
                    while (!stack.isEmpty()) {
                        brr[count++] = stack.peek();
                        brr[count++] = ' ';
                        stack.pop();
                    }
                    stack.push(arr[i]);
                }
            }
        }
        while (!stack.empty()) {
            brr[count++] = stack.peek();
            brr[count++] = ' ';
            stack.pop();
        }
        return brr;
    }

    public static String calculate(char[] brr) {//后缀表达式求得计算结果
        Stack<Double> stack = new Stack();//栈用来进出运算的数字
        for (int i = 0; i < brr.length; ) {
            String number = "";
            if (brr[i] == ' '||brr[i]=='\0') {//空格或为空
                i++;
                continue;
            }
            if (brr[i] >= '0' && brr[i] <= '9') {//数字入栈
                while (brr[i] >= '0' && brr[i] <= '9') {
                    number += brr[i];
                    i++;
```

```
                }
                stack.push(Double.valueOf(number));//字符型转整型入栈
            } else if(brr[i]=='+'||brr[i]=='-'||brr[i]=='*'||brr[i]=='/') {//符号
                double top1 = stack.peek();
                stack.pop();
                double top2 = stack.peek();
                stack.pop();
                double result = operate(top2,top1,brr[i]);//注意 top2 和 top1 的位置
                stack.push(result);
                i++;
            }
        }
        return(stack.peek().toString());
    }

    public static double operate(Double item1, Double item2, char operator) {
        double result = 0;
        switch (operator) {
            case '+':
                result = item1 + item2;
                break;
            case '-':
                result = item1 - item2;
                break;
            case '*':
                result = item1 * item2;
                break;
            case '/':
                result = item1 / item2;
                break;
        }
        return result;
    }

    public static String resu (String exp)
    {
        char brr[] = middleToBack(exp);
        return calculate(brr);
    }
/*
    public static void main(String[] args)
    {
        String exp = "9+(3-1)*3+10/2";
        char[] brr = middleToBack(exp);
```

```
        calculate(brr);
    }
*/
}
```

4.3 textview.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >
    <!-- 设置框内填充颜色 -->
    <solid android:color="#ffffff" />
    <!-- 设置边框宽度和颜色 -->
    <stroke
        android:width="1dip"
        android:color="#000000" />
    <!-- 设置圆角半径 -->
    <corners android:radius="3dp" />
    <!-- 设置边距 -->
    <padding
        android:bottom="5dp"
        android:left="5dp"
        android:right="5dp"
        android:top="5dp" />
    <!-- 设置渐变角度 angle 和渐变颜色 -->
    <gradient
        android:angle="270"
        android:startColor="#87CEFA"
        android:endColor=" #00FA9A"/>
    <!-- 设置各边倒角大小 -->
    <corners
        android:bottomLeftRadius="0dp"
        android:bottomRightRadius="0dp"
        android:topLeftRadius="0dp"
        android:topRightRadius="0dp" />
</shape>
```

4.4 testview2.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >
    <!-- 设置框内填充颜色 -->
    <solid android:color="#ffffff" />
```

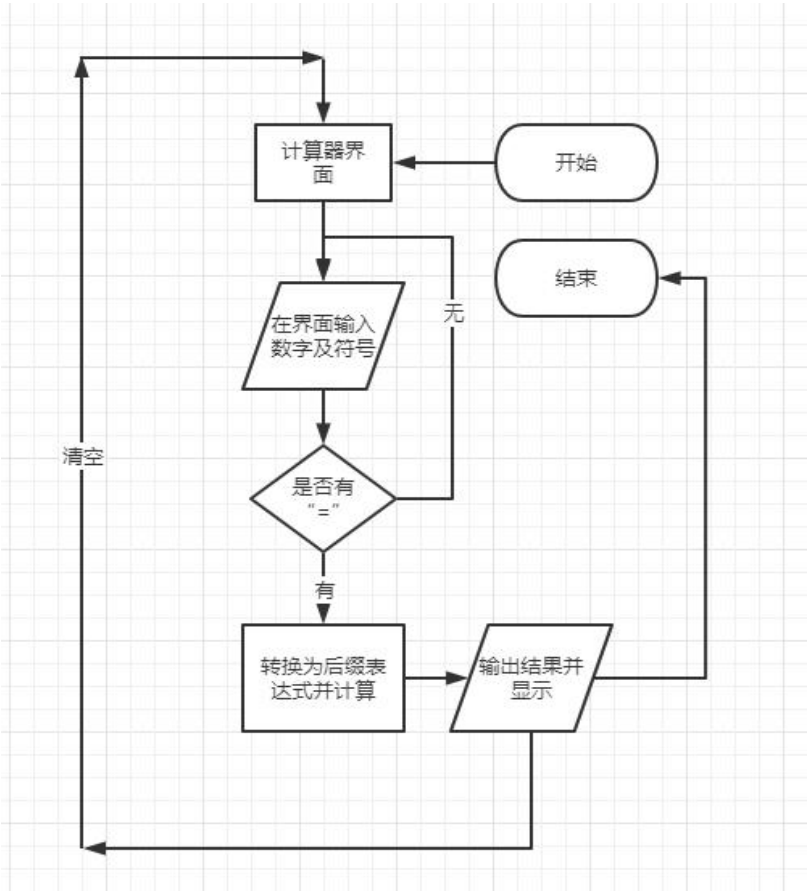
```
    <!-- 设置边框宽度和颜色 -->
    <stroke
        android:width="1dip"
        android:color="#000000" />
    <!-- 设置圆角半径 -->
    <corners android:radius="3dp" />
    <!-- 设置边距 -->
    <padding
        android:bottom="5dp"
        android:left="5dp"
        android:right="5dp"
        android:top="5dp" />
    <!-- 设置渐变角度 angle 和渐变颜色 -->
    <gradient
        android:angle="270"
        android:startColor="#87CEFA"
        android:endColor=" #87CEFA"/>
    <!-- 设置各边倒角大小 -->
    <corners
        android:bottomLeftRadius="0dp"
        android:bottomRightRadius="0dp"
        android:topLeftRadius="0dp"
        android:topRightRadius="0dp" />
</shape>
```

4.5 button.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >
    <!-- 设置框内填充颜色 -->
    <solid android:color="#ffffff" />
    <!-- 设置边框宽度和颜色 -->
    <stroke
        android:width="1dip"
        android:color="#000000" />
    <!-- 设置圆角半径 -->
    <corners android:radius="3dp" />
    <!-- 设置边距 -->
    <padding
        android:bottom="5dp"
        android:left="5dp"
        android:right="5dp"
        android:top="5dp" />
```

```
    <!-- 设置渐变角度 angle 和渐变颜色 -->
    <gradient
        android:angle="270"
        android:startColor="#FA8072"
        android:endColor="#FFEFD5"/>
    <!-- 设置各边倒角大小 -->
    <corners
        android:bottomLeftRadius="150dp"
        android:bottomRightRadius="150dp"
        android:topLeftRadius="150dp"
        android:topRightRadius="150dp" />
</shape>
```

五、软件操作流程



六、难点和解决方案

6.1 计算器需要配置大量按钮，其中对于区域按钮很多的配置是一样的，如果单独对每一个按钮进行大量重复的设置不仅繁琐，而且使代码变得很冗杂。
解决方案：自己新建 drawable 文件，并在所有 button 的 background 中都引用这一个 xml 文件，这样只用修改 drawavle 文件就可以对所有的 button 进行修改

6.2 中缀表达式中判断运算顺序、括号有一定的难度。
解决方案：先将中缀表达式用算法转换为后缀表达式，这样计算就不需要考虑运算顺序和括号问题

6.3 Kotlin 较难编写算法，缺少一些必要的数据结构。
解决方案：将算法部分使用 Java 的类编写，然后在 Kotlin 中调用 java 的类

七、不足之处和今后的设想

本计算器实现的功能较为简单，只能实现见简单的四则运算以及括号，对其他比如说平方、立方、对数等运算还不能处理。另外 Ui 界面不是很美观。

今后要做的改进有加入更多功能和符号比如说 cos,sin，使得计算器功能更加强大，同时要参照 Material Design 风格对计算器重新配色，使计算器更加美观。