

華南師範大學

《移动智能应用开发》

计算器的设计与实现说明文档

软件名称 梦幻计算器

完 成 人 曾伟泽

学 号 20182131119

完成时间 2020 年 10 月 17 日

一、软件内容简介

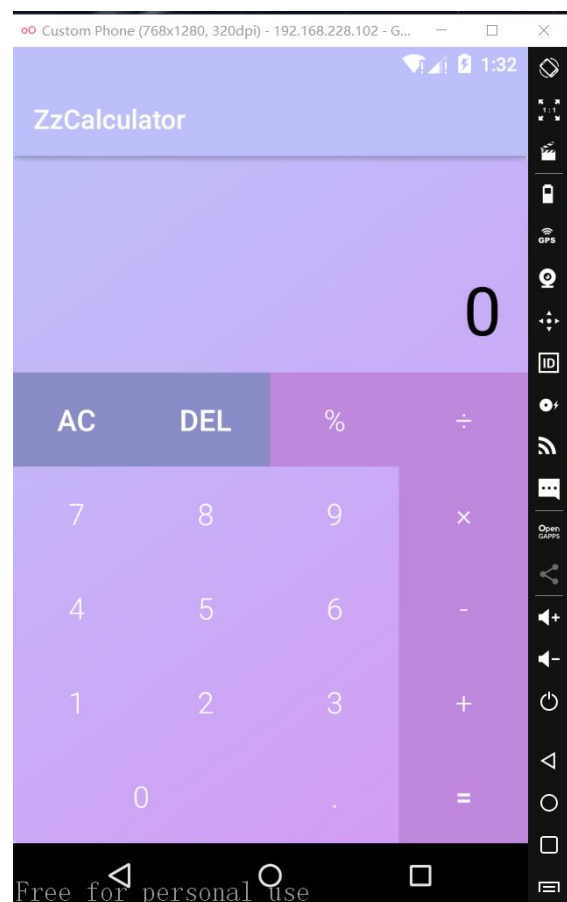
这是一款基于 Android Studio 开发的计算器 APP，Android API 最低支持 Android5.0（API21），适配市面上 94.1%的安卓手机。设计风格我参考了一些计算器 UI 设计大赛，最终决定设计一款紫色渐变色的梦幻色彩风格。

算法设计方面，此计算器支持基础四则运算和求余运算。使用了栈式存储结构，能够把中缀表达式转换为后缀表达式然后进行计算，支持同时多则运算。

综上所述，此计算器 APP 操作简单快捷，UI 简洁梦幻，故称梦幻计算器。

二、界面设计

（一）主界面



采用亮度较高的紫色渐变色作为背景，并通过设置 styles.xml 对状态栏等颜色进行了修改，使之不是突兀的绿色或其他颜色。

(二) 按钮样式



按钮样式

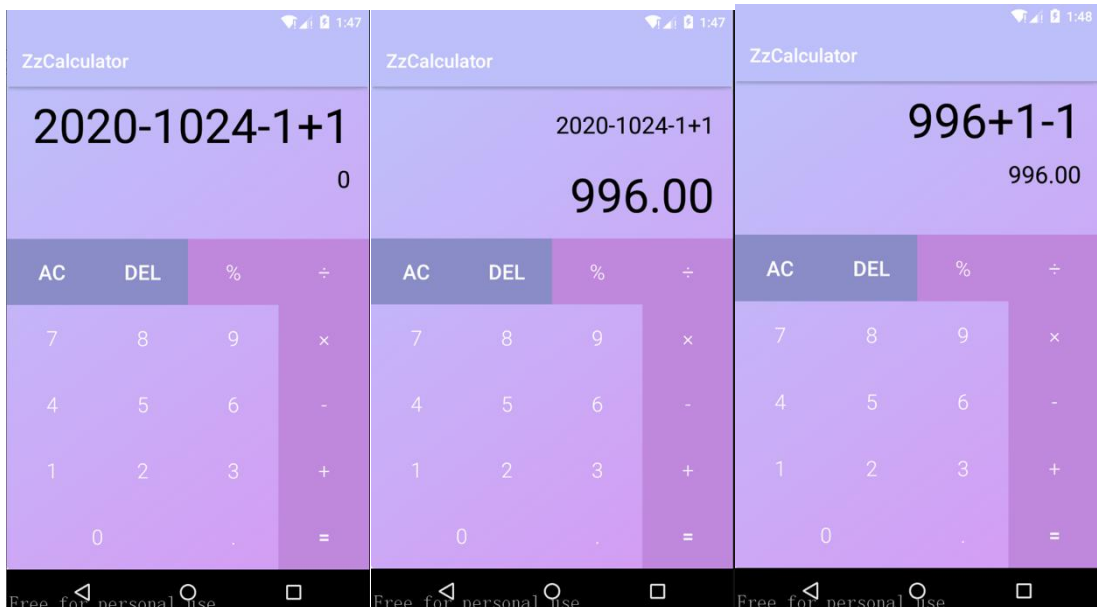
先使用 selector 对按钮做点击响应，然后将其嵌套进入 styles.xml 中

```
<item name="android:background">@drawable/numbtn_set</item>
```

数字键盘初始样式为透明背景，映出渐变色背景；点击后样式为半透明白色，作点中反馈。

功能区与运算符区键盘初始样式为与背景同色系不同明度亮度的紫色，用以与数字键盘区分；点击后样式为透明背景，显现出背景颜色。

(三) 运行(运算)样式



(1)输入数字

(2)点击运算

(3)连续运算

三、代码设计

1. 样式设计 styles.xml

```
<resources>

    <!-- Base application theme. -->

    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">

        <!-- Customize your theme here. -->

        <item name="colorPrimary">@color/colorPrimary</item>

        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>

        <item name="colorAccent">@color/colorAccent</item>

    </style>

    <style name="NumbuttonTheme">

        <item name="android:layout_width">0dp</item>

        <item name="android:layout_height">match_parent</item>

        <item name="android:layout_weight">1</item>

        <item name="android:background">@drawable/numbtn_set</item>

        <item name="android:textSize">25sp</item>

        <item name="android:textStyle">normal</item>

        <item name="android:fontFamily">sans-serif-light</item>

        <item name="android:onClick">clickNumBtn</item>

    </style>

    <style name="OpbuttonTheme">

        <item name="android:layout_width">0dp</item>

        <item name="android:layout_height">match_parent</item>

        <item name="android:layout_weight">1</item>

        <item name="android:background">@drawable/funbtn</item>

        <item name="android:textSize">25sp</item>

        <item name="android:textStyle">normal</item>

    </style>

</resources>
```

```

        <item name="android:fontFamily">sans-serif-light</item>

        <item name="android:onClick">clickNumBtn</item>

    </style>

</resources>

```

1.1 数字键盘按钮 selector 文件 numbtn_set.xml

```

<?xml version="1.0" encoding="utf-8"?>

<selector xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:state_pressed="true"

        android:drawable="@drawable/numbtn"/>

    <item android:state_pressed="false"

        android:drawable="@android:color/transparent"/>

</selector>

```

1.1.1 数字按钮半透明样式文件 numbtn.xml

```

<?xml version="1.0" encoding="utf-8"?>

<shape xmlns:android="http://schemas.android.com/apk/res/android"

    android:shape="rectangle" >

    <solid android:color="#40FFFFFF" />

</shape>

```

1.2 功能按钮及运算符按钮样式文件 funbtn.xml

```

<?xml version="1.0" encoding="utf-8"?>

<selector xmlns:android="http://schemas.android.com/apk/res/android" >

    <!-- 点击按钮 -->

    <item android:state_pressed="true">

        <shape >

            <!-- 填充,颜色 -->

            <solid android:color="@android:color/transparent"/>

            android:text

```

```

        </shape>

    </item>

    <!-- 按钮正常状态,未点击 -->

    <item >

        <shape >

            <!-- 填充,颜色 -->

            <solid android:color="#C088DD"/>

        </shape>

    </item>

</selector>

```

2. 渐变色背景样式文件 backgroundshape.xml

```

<?xml version="1.0" encoding="utf-8"?>

<shape xmlns:android="http://schemas.android.com/apk/res/android"

    android:shape="rectangle">

    <gradient

        android:angle="135"

        android:startColor="#D59AF3"

        android:endColor="#BDBFFA"

    />

    <corners android:radius="4dp"/>

</shape>

```

3. 布局设计文件 activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:orientation="vertical"

```

```

        android:background="@drawable/backgroundshape"

    >
    <EditText
        android:id="@+id/inputTextView"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:focusable="false"
        android:inputType="text"
        android:lines="1"
        android:layout_weight="1"
        android:gravity="center_vertical|right"
        android:paddingRight="20dp"
        android:background="@null"
        android:textColor="#000000"
        android:textSize="50sp"
    />

    <TextView android:layout_width="match_parent" android:layout_height="0dp"
        android:id="@+id/outputTextView"
        android:layout_weight="1"
        android:gravity="right"
        android:textColor="#000000"
        android:text="0"
        android:paddingRight="20dp"
        android:textSize="25sp"/>

    <TableLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TableRow
            android:layout_width="match_parent"

```

```
android:layout_height="wrap_content">
```

```
<Button
```

```
    android:layout_width="0dp"
```

```
    android:layout_weight="1"
```

```
    android:layout_height="70dp"
```

```
    android:textSize="22sp"
```

```
    android:text="AC"
```

```
    android:textColor="#ffffff"
```

```
    android:background="@drawable/buttonshape"
```

```
    android:id="@+id/ac_btn" />
```

```
<Button
```

```
    android:layout_width="0dp"
```

```
    android:layout_weight="1"
```

```
    android:layout_height="70dp"
```

```
    android:textSize="22sp"
```

```
    android:text="Del"
```

```
    android:textColor="#ffffff"
```

```
    android:background="@drawable/buttonshape"
```

```
    android:id="@+id/delete_btn" />
```

```
<Button
```

```
    android:layout_width="0dp"
```

```
    android:layout_weight="1"
```

```
    android:layout_height="70dp"
```

```
    android:textSize="22sp"
```

```
    android:text="%"
```

```
    android:textColor="#ffffff"
```

```
    style="@style/OpbuttonTheme"
```



```
        android:id="@+id/percent_btn" />
```

```
<Button
```

```
    android:layout_width="0dp"
```

```
    android:layout_weight="1"
```

```
    android:layout_height="70dp"
```

```
    android:textSize="22sp"
```

```
    android:text="÷"
```

```
    android:textColor="#ffffff"
```

```
    style="@style/OpbuttonTheme"
```

```
    android:id="@+id/divide_btn" />
```

```
</TableRow>
```

```
<TableRow
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content">
```

```
<Button
```

```
    android:layout_width="0dp"
```

```
    android:layout_weight="1"
```

```
    android:layout_height="70dp"
```

```
    android:textSize="22sp"
```

```
    android:text="7"
```

```
    android:textColor="#ffffff"
```

```
    style="@style/NumbuttonTheme"
```

```
    android:id="@+id/num_seven" />
```

```
<Button
```

```
    android:layout_width="0dp"
```

```
    android:layout_weight="1"
```

```
        android:layout_height="70dp"

        android:textSize="22sp"

        android:text="8"

        android:textColor="#ffffff"

        style="@style/NumbuttonTheme"

        android:id="@+id/num_eight" />
```

<Button

```
        android:layout_width="0dp"

        android:layout_weight="1"

        android:layout_height="70dp"

        android:textSize="22sp"

        android:text="9"

        android:textColor="#ffffff"

        style="@style/NumbuttonTheme"

        android:id="@+id/num_nine" />
```

<Button

```
        android:layout_width="0dp"

        android:layout_weight="1"

        android:layout_height="70dp"

        android:textSize="22sp"

        android:text="x"

        android:textColor="#ffffff"

        style="@style/OpbuttonTheme"

        android:id="@+id/multiply_btn" />
```

</TableRow>

<TableRow

```
        android:layout_width="match_parent"
```

```
android:layout_height="wrap_content">
```

```
<Button
```

```
    android:layout_width="0dp"

    android:layout_weight="1"

    android:layout_height="70dp"

    android:textSize="22sp"

    android:text="4"

    android:textColor="#ffffff"

    style="@style/NumbuttonTheme"

    android:id="@+id/num_four" />
```

```
<Button
```

```
    android:layout_width="0dp"

    android:layout_weight="1"

    android:layout_height="70dp"

    android:textSize="22sp"

    android:text="5"

    android:textColor="#ffffff"

    style="@style/NumbuttonTheme"

    android:id="@+id/num_five" />
```

```
<Button
```

```
    android:layout_width="0dp"

    android:layout_weight="1"

    android:layout_height="70dp"

    android:textSize="22sp"

    android:text="6"

    android:textColor="#ffffff"

    style="@style/NumbuttonTheme"
```

```
        android:id="@+id/num_six" />
```

```
    <Button
```

```
        android:layout_width="0dp"
```

```
        android:layout_weight="1"
```

```
        android:layout_height="70dp"
```

```
        android:textSize="22sp"
```

```
        android:text="- "
```

```
        android:textColor="#ffffff"
```

```
        style="@style/OpbuttonTheme"
```

```
        android:id="@+id/subtract_btn" />
```

```
</TableRow>
```

```
<TableRow
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content">
```

```
    <Button
```

```
        android:layout_width="0dp"
```

```
        android:layout_weight="1"
```

```
        android:layout_height="70dp"
```

```
        android:textSize="22sp"
```

```
        android:text="1"
```

```
        android:textColor="#ffffff"
```

```
        style="@style/NumbuttonTheme"
```

```
        android:id="@+id/num_one" />
```

```
    <Button
```

```
        android:layout_width="0dp"
```

```
        android:layout_weight="1"
```

```
        android:layout_height="70dp"

        android:textSize="22sp"

        android:text="2"

        android:textColor="#ffffff"

        style="@style/NumbuttonTheme"

        android:id="@+id/num_two" />
```

<Button

```
        android:layout_width="0dp"

        android:layout_weight="1"

        android:layout_height="70dp"

        android:textSize="22sp"

        android:text="3"

        android:textColor="#ffffff"

        style="@style/NumbuttonTheme"

        android:id="@+id/num_three" />
```

<Button

```
        android:layout_width="0dp"

        android:layout_weight="1"

        android:layout_height="70dp"

        android:textSize="22sp"

        android:text="+"

        android:textColor="#ffffff"

        style="@style/OpbuttonTheme"

        android:id="@+id/plus_btn" />
```

</TableRow>

<TableRow

```
        android:layout_width="match_parent"
```

```
android:layout_height="wrap_content">
```

```
<Button
```

```
    android:layout_width="0dp"

    android:layout_weight="2"

    android:layout_height="70dp"

    android:textSize="22sp"

    android:text="0"

    android:textColor="#ffffff"

    style="@style/NumbuttonTheme"

    android:id="@+id/num_zero" />
```

```
<Button
```

```
    android:layout_width="0dp"

    android:layout_weight="1"

    android:layout_height="70dp"

    android:textSize="22sp"

    android:text="."

    android:textColor="#ffffff"

    style="@style/NumbuttonTheme"

    android:id="@+id/dot_btn" />
```

```
<Button
```

```
    android:layout_width="0dp"

    android:layout_weight="1"

    android:layout_height="70dp"

    android:textSize="22sp"

    android:text=""

    android:textColor="#ffffff"

    android:background="@drawable/funbtn"
```

```

                android:id="@+id/equal_btn" />

            </TableRow>

        </TableLayout>

    </LinearLayout>

```

4. MainActivity.kt

```

package com.example.zzcalculator

import android.os.Bundle
import android.view.View
import android.widget.Button
import androidx.appcompat.app.AppCompatActivity
import kotlinx.android.synthetic.main.activity_main.*
import java.util.*
import android.textEditable
import java.text.DecimalFormat

class MainActivity : AppCompatActivity() {

    var numberStack = Stack<Double>() //存放数字的栈

    var symbolStack = Stack<Char>() //存放运算符的栈

    var clickOperator = false //防止出现重复运算符

    var clickPoint = true

    var clickEqt = false

    override fun onCreate(savedInstanceState: Bundle?) {

```

```
super.onCreate(savedInstanceState)

setContentView(R.layout.activity_main)

equal_btn.setOnClickListener {

    var ss = inputTextView.text.toString()

    if (ss!=""){

        clickEq=true

        clickPoint=true

        ss += '='

        inputTextView.textSize = 25F

        outputTextView.textSize = 50F

        if(!isStandard(ss)){

            outputTextView.text="Error!"

        }

        else{

            outputTextView.text=calculate(ss)

            //如果文本过长，字体变小

            val tp = outputTextView.paint

            tp.textSize = outputTextView.textSize

            val tvw = tp.measureText(outputTextView.text.toString())

            if (outputTextView.width < tvw) {

                outputTextView.textSize = 30F

            }

        }

    }

}
```



```

ac_btn.setOnClickListener {
    inputTextView.setText("")
    outputTextView.text="0"
    clickOperator=false
    clickPoint=true
    clickEq=false
}

delete_btn.setOnClickListener {
    var ss = inputTextView.text.toString()

    try{
        val lastCh=ss[ss.lastIndex]

        if(ss!=""){
            if (lastCh == '+' || lastCh == '-' || lastCh == 'x' || lastCh ==
'÷' || lastCh == '%') {
                clickOperator = true
            }

            if (lastCh == '.') {
                clickPoint = true
            }

            ss = ss.substring(0, ss.lastIndex)
            inputTextView.setText(ss)
        }
    }catch (e:StringIndexOutOfBoundsException){
        print(e.stackTrace)
    }
}

}

fun clickNumBtn(v:View?){

```

```
when(v?.id){  
  
    R.id.plus_btn->{  
  
        if(clickOperator){  
  
            clickOpBtn()  
  
            inputTextView.append("+")  
  
            clickOperator = false  
  
            clickPoint = true  
  
        }  
    }  
  
    R.id.subtract_btn->{  
  
        if(clickOperator){  
  
            clickOpBtn()  
  
            inputTextView.append("-")  
  
            clickOperator = false  
  
            clickPoint = true  
  
        }  
    }  
  
    R.id.multiply_btn->{  
  
        if(clickOperator){  
  
            clickOpBtn()  
  
            inputTextView.append("×")  
  
            clickOperator = false  
  
            clickPoint = true  
  
        }  
    }  
  
    R.id.divide_btn->{  
  
        if(clickOperator){  
  
            clickOpBtn()  
  
            inputTextView.append("÷")  
  
            clickOperator = false
```

```
        clickPoint = true
    }
}

R.id.percent_btn->{
    if(clickOperator){
        clickOpBtn()
        inputTextView.append("%")
        clickOperator = false
        clickPoint = true
    }
}

R.id.dot_btn->{
    if(clickOperator){
        clickOpBtn()
        inputTextView.append(".")
        clickPoint=false
    }
}

R.id.num_zero->{
    clickNumBtn()
    inputTextView.append("0")
    clickOperator = true
}

R.id.num_one->{
    clickNumBtn()
    inputTextView.append("1")
    clickOperator = true
}

R.id.num_two->{
    clickNumBtn()
```

```
        inputTextView.append("2")

        clickOperator = true
    }

    R.id.num_three->{

        clickNumBtn()

        inputTextView.append("3")

        clickOperator = true
    }

    R.id.num_four->{

        clickNumBtn()

        inputTextView.append("4")

        clickOperator = true
    }

    R.id.num_five->{

        clickNumBtn()

        inputTextView.append("5")

        clickOperator = true
    }

    R.id.num_six->{

        clickNumBtn()

        inputTextView.append("6")

        clickOperator = true
    }

    R.id.num_seven->{

        clickNumBtn()

        inputTextView.append("7")

        clickOperator = true
    }

    R.id.num_eight->{

        clickNumBtn()
```

```

        inputTextView.append("8")

        clickOperator = true
    }

    R.id.num_nine->{

        clickNumBtn()

        inputTextView.append("9")

        clickOperator = true
    }
}

fun clickOpBtn(){

    outputTextView.textSize = 25F

    inputTextView.textSize = 50F

    if(clickEq){

        inputTextView.setText(outputTextView.text)

        clickEq = false
    }

}

fun clickNumBtn() {

    outputTextView.textSize = 25F

    inputTextView.textSize = 50F

    if (clickEq) {

        inputTextView.setText("")

        clickEq = false
    }

}

fun calculate(numStr: String): String {

    symbolStack.push('=')

    var temp = StringBuffer()

    val length = numStr.length - 1

```

```

for (i in 0..length) {

    val ch = numStr[i]

    if (isNumber(ch) || ch == '.' || ch == 'E') {

        temp.append(ch)

    } else {

        try {

            if (temp[0] == '.') { //将 .123 转为 0.123

                temp.insert(0, '0')

            }

            if (temp[temp.LastIndex] == '.') {

                return "Error!"

            }

        } catch (e: StringIndexOutOfBoundsException) {

            print(e.stackTrace)

        }

        val tempStr = temp.toString()

        if (tempStr.isNotEmpty()) {

            val num = tempStr.toDouble()

            numberStack.push(num)

            temp = StringBuffer() //重置

        }

        /**

         * if icp(ch) > isp(top) , 令ch 进栈

         * if icp(ch) < isp(top) , top 退栈运算

         * if icp(ch) == isp(top) , top 退栈但不操作

         */

        var top = symbolStack.peek()

```

```
if (icp(ch) > isp(top)) {  
    symbolStack.push(ch)  
} else {  
    while (icp(ch) < isp(top)) {  
        val b = numberStack.pop()  
        val a = numberStack.pop()  
  
        when (symbolStack.pop()) {  
            '+' -> numberStack.push(a + b)  
            '-' -> numberStack.push(a - b)  
            'x' -> numberStack.push(a * b)  
            '÷' -> {  
                if (b == 0.0) {  
                    return "除数不能为零"  
                }  
                numberStack.push(a / b)  
            }  
            '%' -> numberStack.push(a % b)  
        }  
        top = symbolStack.peek()  
    }  
    if (icp(ch) > isp(top)) {  
        symbolStack.push(ch)  
    }  
    if (icp(ch) == isp(top)) {  
        symbolStack.pop()  
    }  
}
```

```

    }

}

val df=DecimalFormat("#####0.00")

var result = df.format(numberStack.pop())

return result.toString()
}

fun isStandard(numStr: String): Boolean {

    //等号前不是数字，如 1+ 或者 ")"

    if (!isNumber(numStr[numStr.LastIndex - 1]) && numStr[numStr.LastIndex - 1] != '('))

        return false

    val stack = Stack<Char>()    // 用来保存括号，检查左右括号是否匹配

    val length = numStr.length - 1

    for (i in 0..length) {

        val n = numStr[i]

        if (!isNumber(n) && n != '+' && n != '-' && n != 'x' && n != '÷' && n != '%' && n != '(' && n != ')' && n != '.' && n != 'E' && n != '=')

            return false

        // 将左括号压栈，用来给后面的右括号进行匹配

        if (n == '(') {

            stack.push(n)

        }

        if (n == ')') {

            if (stack.isEmpty() || stack.pop() != '(')

```



```

        return false
    }
}

// 可能会有缺少右括号的情况

if (!stack.isEmpty())
    return false

return true
}

//判断字符是否是 0-9 的数字

private fun isNumber(num: Char): Boolean {
    if (num in '0'..'9')
        return true
    return false
}

//栈内优先数

private fun isp(ch: Char): Int {
    when (ch) {
        '=' -> return 0
        '(' -> return 1
        '×', '÷', '%' -> return 5
        '+', '-' -> return 3
        ')' -> return 6
    }
    return -1
}

```

```

//栈外优先数

private fun icp(ch: Char): Int {

    when (ch) {

        '=' -> return 0

        '(' -> return 6

        '×', '÷', '%' -> return 4

        '+', '-' -> return 2

        ')' -> return 1

    }

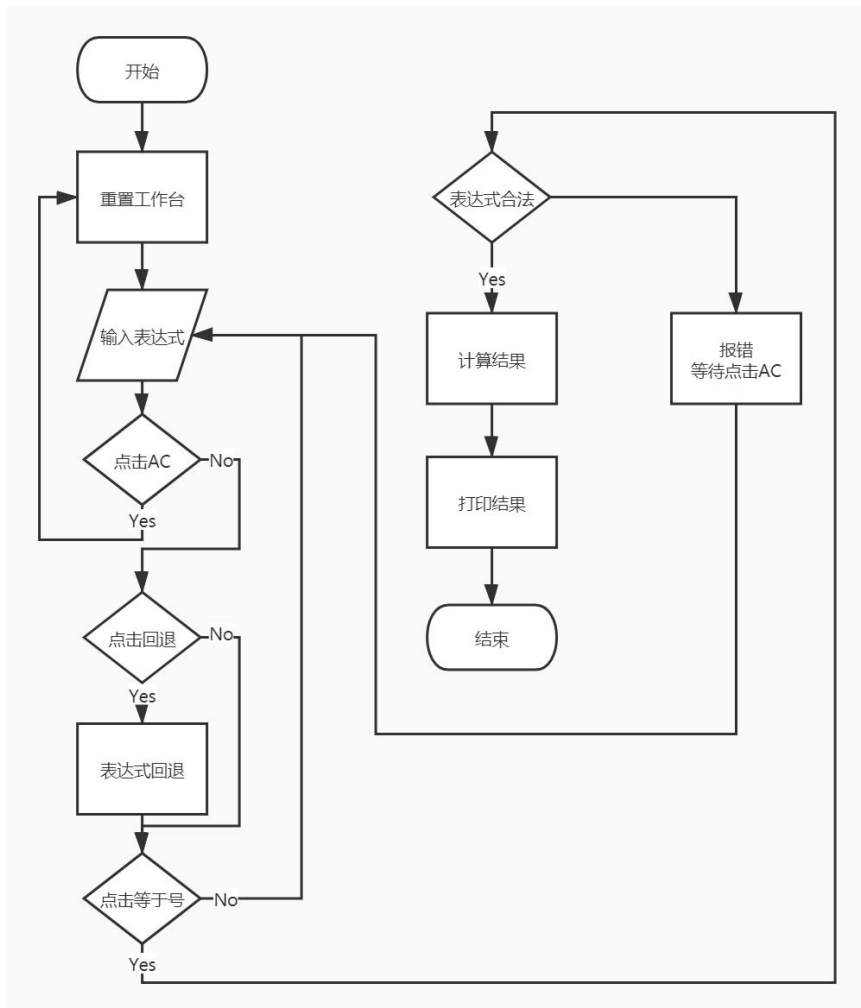
    return -1

}

}

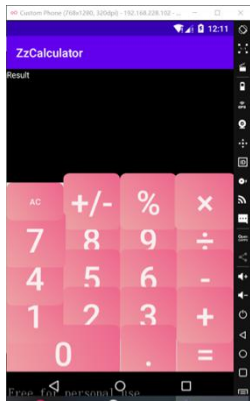
```

四、软件操作流程



五、难点（或遇到的问题）和解决方案

开始时选用约束布局，但是无论怎么约束、设置辅助线等都没有得到理想的布局，某些情况下按钮还会乱



后来参考其他计算器 app 的布局，选用了如今线性布局嵌套 table row 的布局。

六、不足之处

- 1.按钮样式嵌套文件过多，可以精简
- 2.%按钮进行了求余运算而不是日常生活中更常见的百分比运算
- 3.运算表达式过长时会隐藏先输入的数据而不是文本缩小换行以完整显示

七、今后的设想

- 1.完善计算器功能

添加运算括号的按钮，实际算法中包含了括号的处理，但是布局设计时遗落了输入括号的按钮

- 2.修改%按钮功能

将其由求余运算改成百分比运算

3.开发自定义功能

开放用户修改背景颜色的接口，使用户可以自行修改计算器的配色