

# 计算器的设计与实现说明文档

完成人：梁诺明 学号：20192131031

## 一、软件名称

简易计算器

## 二、软件内容简介

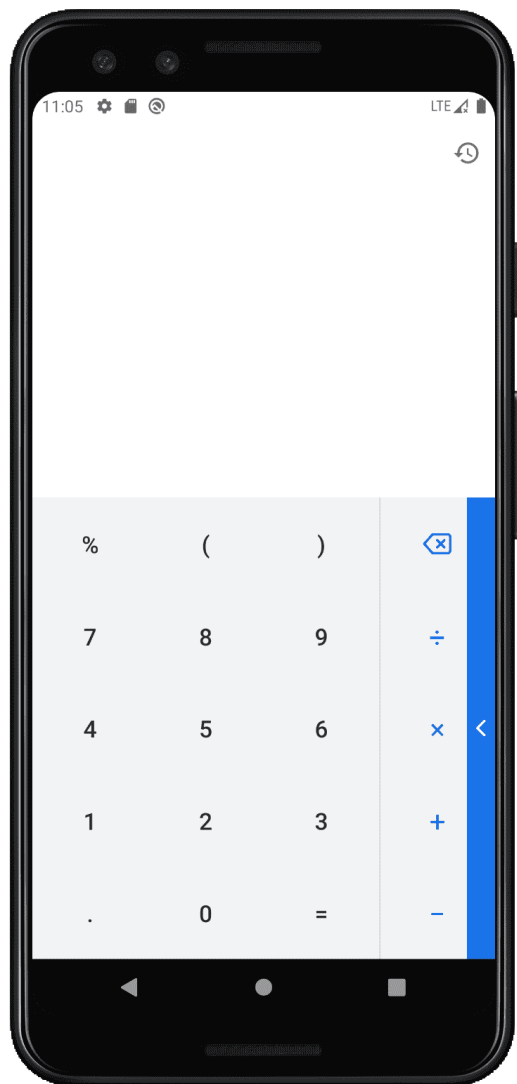
这是一个用kotlin编写的较为简易的科学计算器，软件界面参考了Android原生系统的计算器样式。可以处理常见的科学计算，有历史记录功能，可以找回曾经计算过的表达式。

历史记录功能：用 `List<HistoryEntity>` 显示存储在本地的历史记录数据，数据结构如下

```
1  @Entity(tableName = "history")
2  data class HistoryEntity(
3      @PrimaryKey(autoGenerate = false) @ColumnInfo(name = "expression") val expression: String,
4      @ColumnInfo(name = "result") val result: String,
5      @ColumnInfo(name = "date") val date: Long
6  )
```

## 三、界面设计

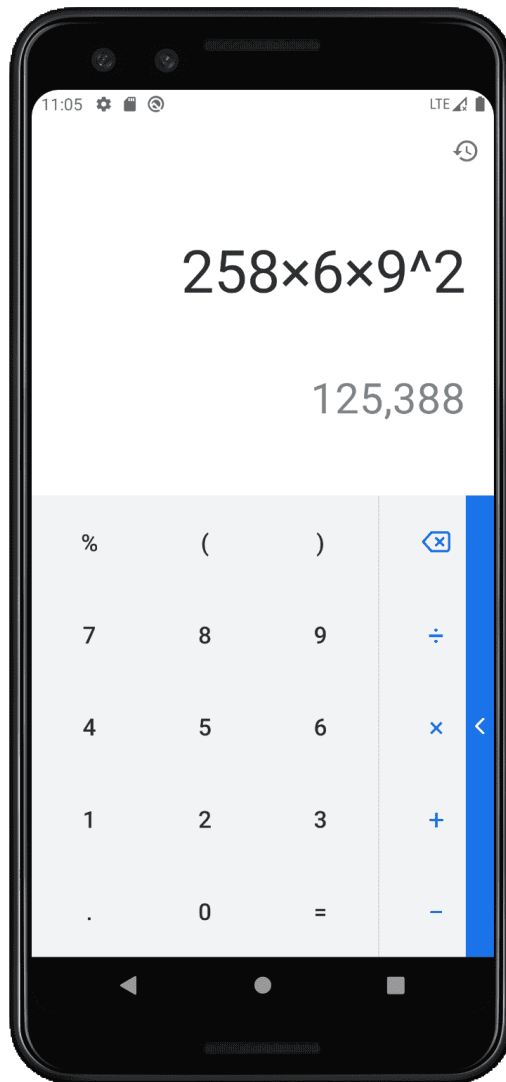
1.主界面



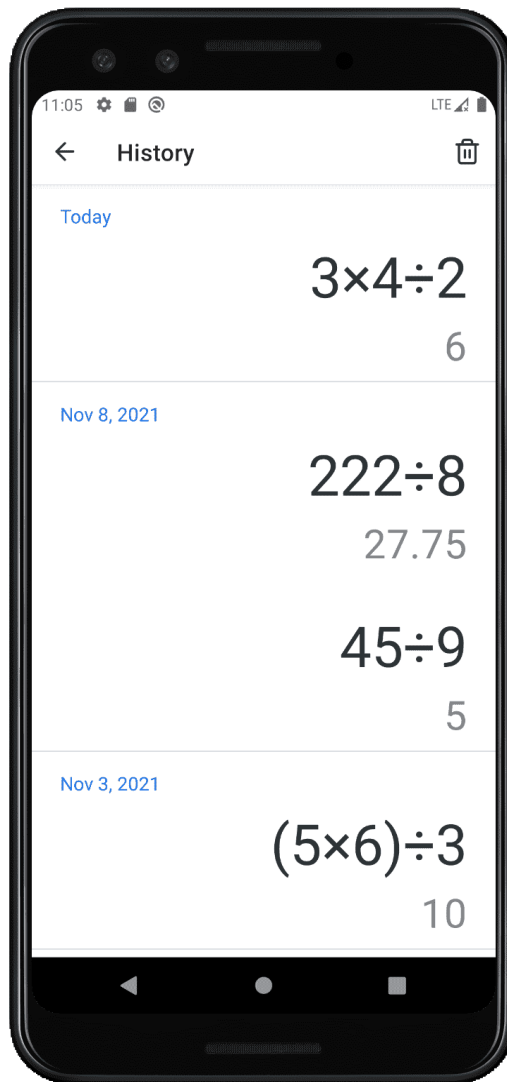
2.侧滑科学计算面板



### 3.计算结果展示



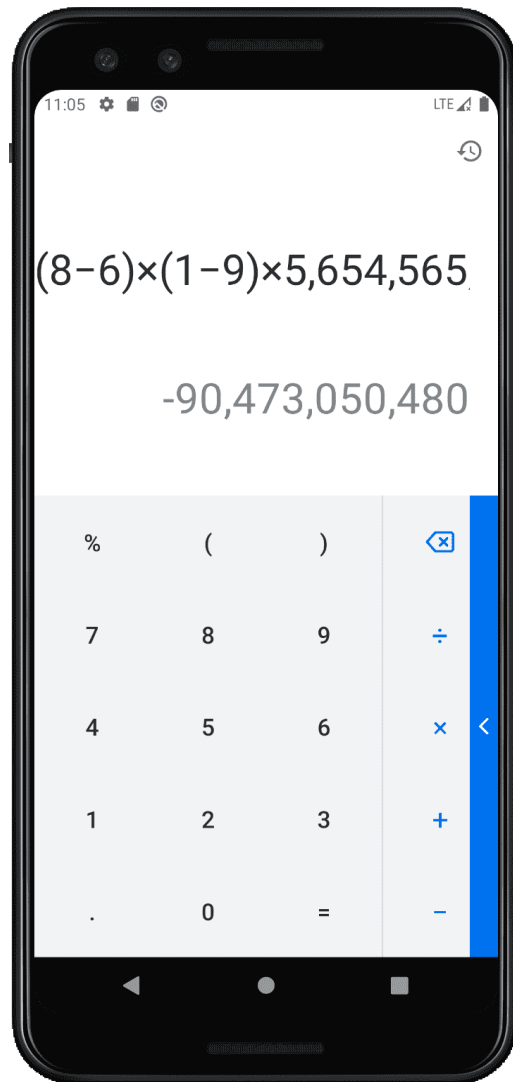
#### 4.历史记录功能

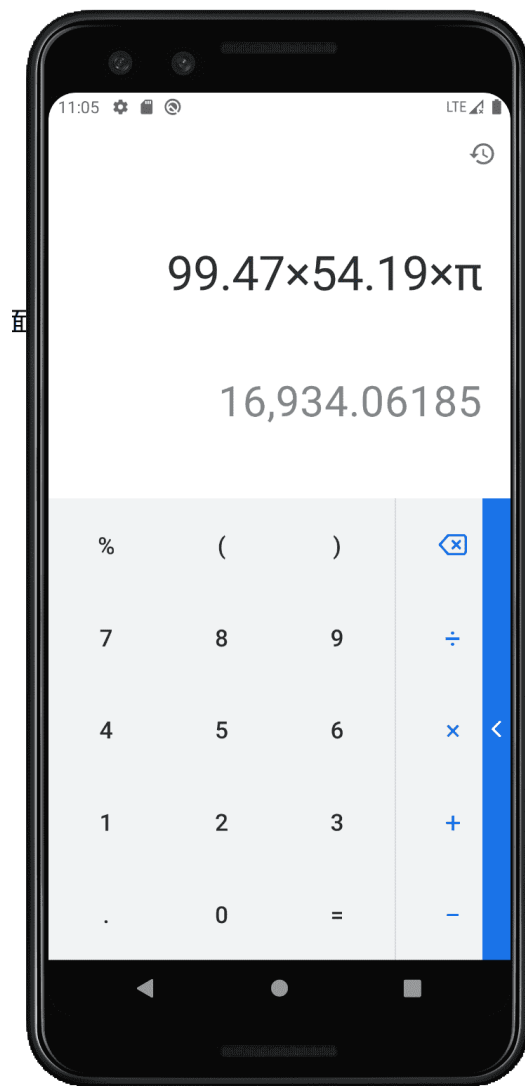


## 5. 括号运算



6.表达式过长自动缩小





## 四、关键代码

### 一、布局文件

1.activity\_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context=".ui.main.MainActivity">
9      <com.google.android.material.appbar.AppBarLayout
```



```

10         android:id="@+id/appBar"
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:background="?attr/colorSurface"
14         app:elevation="0dp"
15         app:layout_constraintEnd_toEndOf="parent"
16         app:layout_constraintStart_toStartOf="parent"
17         app:layout_constraintTop_toTopOf="parent">
18
19         <com.google.android.material.appbar.MaterialToolbar
20             android:id="@+id/toolbar"
21             android:layout_width="match_parent"
22             android:layout_height="?actionBarSize"
23             app:menu="@menu/main_menu"
24             app:title=" " />
25     </com.google.android.material.appbar.AppBarLayout>
26
27     <include
28         android:id="@+id/resultPad"
29         layout="@layout/result_pad"
30         android:layout_width="match_parent"
31         android:layout_height="0dp"
32         app:layout_constraintBottom_toTopOf="@+id/guideline"
33         app:layout_constraintEnd_toEndOf="parent"
34         app:layout_constraintStart_toStartOf="parent"
35         app:layout_constraintTop_toBottomOf="@+id/appBar" />
36
37     <com.gigaworks.tech.calculator.ui.view.CalculatorPadViewPager
38         android:id="@+id/calculatorPadViewPager"
39         android:layout_width="match_parent"
40         android:layout_height="0dp"
41         app:layout_constraintBottom_toBottomOf="parent"
42         app:layout_constraintEnd_toEndOf="parent"
43         app:layout_constraintStart_toStartOf="parent"
44         app:layout_constraintTop_toTopOf="@+id/guideline">
45
46         <include
47             android:id="@+id/numPad"
48             layout="@layout/num_pad" />
49
50         <include
51             android:id="@+id/scientificPad"
52             layout="@layout/scientific_pad" />
53
54     </com.gigaworks.tech.calculator.ui.view.CalculatorPadViewPager>
55
56     <androidx.constraintlayout.widget.Guideline

```

```

57         android:id="@+id/guideline"
58         android:layout_width="wrap_content"
59         android:layout_height="wrap_content"
60         android:orientation="horizontal"
61         app:layout_constraintGuide_percent="0.45" />
62
63     <View
64         android:id="@+id/clearView"
65         android:layout_width="0dp"
66         android:layout_height="0dp"
67         android:elevation="4dp"
68         android:background="?attr/clearColor"
69         android:visibility="invisible"
70         app:layout_constraintBottom_toTopOf="@+id/guideline"
71         app:layout_constraintEnd_toEndOf="parent"
72         app:layout_constraintStart_toStartOf="parent"
73         app:layout_constraintTop_toTopOf="@+id/appBar" />
74
75 </androidx.constraintlayout.widget.ConstraintLayout>

```

## 2.num\_pad.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      xmlns:tools="http://schemas.android.com/tools">
8      <androidx.constraintlayout.widget.ConstraintLayout
9          android:id="@+id/firstRow"
10         android:layout_width="match_parent"
11         android:layout_height="0dp"
12         app:layout_constraintBottom_toTopOf="@id/secondRow"
13         app:layout_constraintEnd_toEndOf="parent"
14         app:layout_constraintStart_toStartOf="parent"
15         app:layout_constraintTop_toTopOf="parent">
16
17         <com.google.android.material.button.MaterialButton
18             android:id="@+id/percent"
19             style="@style/Widget.CalculatorPlus.NumPad.Primary"
20             android:layout_width="0dp"
21             android:layout_height="match_parent"
22             android:text="%"

```

```
23         android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
24         app:layout_constraintEnd_toStartOf="@id/openBracket"
25         app:layout_constraintStart_toStartOf="parent"
26         app:layout_constraintTop_toTopOf="parent"
27         tools:ignore="HardcodedText" />
28
29     <com.google.android.material.button.MaterialButton
30         android:id="@+id/openBracket"
31         style="@style/Widget.CalculatorPlus.NumPad.Primary"
32         android:layout_width="0dp"
33         android:layout_height="match_parent"
34         android:text="("
35         android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
36         app:layout_constraintEnd_toStartOf="@id/closeBracket"
37         app:layout_constraintStart_toEndOf="@id/percent"
38         app:layout_constraintTop_toTopOf="parent"
39         tools:ignore="HardcodedText" />
40
41     <com.google.android.material.button.MaterialButton
42         android:id="@+id/closeBracket"
43         style="@style/Widget.CalculatorPlus.NumPad.Primary"
44         android:layout_width="0dp"
45         android:layout_height="match_parent"
46         android:text=")"
47         android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
48         app:layout_constraintEnd_toStartOf="@id/delete"
49         app:layout_constraintStart_toEndOf="@id/openBracket"
50         app:layout_constraintTop_toTopOf="parent"
51         tools:ignore="HardcodedText" />
52
53     <ImageButton
54         android:id="@+id/delete"
55         style="@style/Widget.CalculatorPlus.NumPad.Primary.IconOnly"
56         android:layout_width="0dp"
57         android:layout_height="match_parent"
58         android:scaleType="fitCenter"
59         android:src="@drawable/delete"
60         app:layout_constraintEnd_toEndOf="parent"
61         app:layout_constraintStart_toEndOf="@id/closeBracket"
62         app:layout_constraintTop_toTopOf="parent"
63         android:contentDescription="@string/delete_button" />
64
65     <View
66         android:layout_width="1dp"
67         android:layout_height="match_parent"
68         android:background="?attr/border"
69         app:layout_constraintTop_toTopOf="parent"
```

```

70         app:layout_constraintBottom_toBottomOf="parent"
71         app:layout_constraintStart_toEndOf="@id/closeBracket"
72         app:layout_constraintEnd_toStartOf="@id/delete"/>
73
74     </androidx.constraintlayout.widget.ConstraintLayout>
75
76     <androidx.constraintlayout.widget.ConstraintLayout
77         android:id="@+id/secondRow"
78         android:layout_width="match_parent"
79         android:layout_height="0dp"
80         app:layout_constraintBottom_toTopOf="@id/thirdRow"
81         app:layout_constraintEnd_toEndOf="parent"
82         app:layout_constraintStart_toStartOf="parent"
83         app:layout_constraintTop_toBottomOf="@id/firstRow">
84
85         <com.google.android.material.button.MaterialButton
86             android:id="@+id/seven"
87             style="@style/Widget.CalculatorPlus.NumPad.Primary"
88             android:layout_width="0dp"
89             android:layout_height="match_parent"
90             android:text="7"
91             android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
92             app:layout_constraintEnd_toStartOf="@id/eight"
93             app:layout_constraintStart_toStartOf="parent"
94             app:layout_constraintTop_toTopOf="parent"
95             tools:ignore="HardcodedText" />
96
97         <com.google.android.material.button.MaterialButton
98             android:id="@+id/eight"
99             style="@style/Widget.CalculatorPlus.NumPad.Primary"
100            android:layout_width="0dp"
101            android:layout_height="match_parent"
102            android:text="8"
103            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
104            app:layout_constraintEnd_toStartOf="@id/nine"
105            app:layout_constraintStart_toEndOf="@id/seven"
106            app:layout_constraintTop_toTopOf="parent"
107            tools:ignore="HardcodedText" />
108
109        <com.google.android.material.button.MaterialButton
110            android:id="@+id/nine"
111            style="@style/Widget.CalculatorPlus.NumPad.Primary"
112            android:layout_width="0dp"
113            android:layout_height="match_parent"
114            android:text="9"
115            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
116            app:layout_constraintEnd_toStartOf="@id/divide"

```

```

16         app:layout_constraintStart_toEndOf="@id/eight"
17         app:layout_constraintTop_toTopOf="parent"
18         tools:ignore="HardcodedText" />
19
20         <com.google.android.material.button.MaterialButton
21             android:id="@+id/divide"
22             style="@style/Widget.CalculatorPlus.NumPad.Primary"
23             android:layout_width="0dp"
24             android:layout_height="match_parent"
25             android:text="\u00F7"
26             android:textColor="?attr/operatorBtnColor"
27             android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle2"
28             app:layout_constraintEnd_toEndOf="parent"
29             app:layout_constraintStart_toEndOf="@id/nine"
30             app:layout_constraintTop_toTopOf="parent"
31             tools:ignore="HardcodedText" />
32
33     <View
34         android:layout_width="1dp"
35         android:layout_height="match_parent"
36         android:background="?attr/border"
37         app:layout_constraintTop_toTopOf="parent"
38         app:layout_constraintBottom_toBottomOf="parent"
39         app:layout_constraintStart_toEndOf="@id/nine"
40         app:layout_constraintEnd_toStartOf="@id/divide"/>
41
42 </androidx.constraintlayout.widget.ConstraintLayout>
43
44 <androidx.constraintlayout.widget.ConstraintLayout
45     android:id="@+id/thirdRow"
46     android:layout_width="match_parent"
47     android:layout_height="0dp"
48     app:layout_constraintBottom_toTopOf="@id/fourthRow"
49     app:layout_constraintEnd_toEndOf="parent"
50     app:layout_constraintStart_toStartOf="parent"
51     app:layout_constraintTop_toBottomOf="@id/secondRow">
52
53     <com.google.android.material.button.MaterialButton
54         android:id="@+id/four"
55         style="@style/Widget.CalculatorPlus.NumPad.Primary"
56         android:layout_width="0dp"
57         android:layout_height="match_parent"
58         android:text="4"
59         android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
60         app:layout_constraintEnd_toStartOf="@id/five"
61         app:layout_constraintStart_toStartOf="parent"
62         app:layout_constraintTop_toTopOf="parent"

```

```

13         tools:ignore="HardcodedText" />
14
15         <com.google.android.material.button.MaterialButton
16             android:id="@+id/five"
17             style="@style/Widget.CalculatorPlus.NumPad.Primary"
18             android:layout_width="0dp"
19             android:layout_height="match_parent"
20             android:text="5"
21             android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
22             app:layout_constraintEnd_toStartOf="@id/six"
23             app:layout_constraintStart_toEndOf="@id/four"
24             app:layout_constraintTop_toTopOf="parent"
25             tools:ignore="HardcodedText" />
26
27         <com.google.android.material.button.MaterialButton
28             android:id="@+id/six"
29             style="@style/Widget.CalculatorPlus.NumPad.Primary"
30             android:layout_width="0dp"
31             android:layout_height="match_parent"
32             android:text="6"
33             android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
34             app:layout_constraintEnd_toStartOf="@id/multiply"
35             app:layout_constraintStart_toEndOf="@id/five"
36             app:layout_constraintTop_toTopOf="parent"
37             tools:ignore="HardcodedText" />
38
39         <com.google.android.material.button.MaterialButton
40             android:id="@+id/multiply"
41             style="@style/Widget.CalculatorPlus.NumPad.Primary"
42             android:layout_width="0dp"
43             android:layout_height="match_parent"
44             android:text="\u00D7"
45             android:textColor="?attr/operatorBtnColor"
46             android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle2"
47             app:layout_constraintEnd_toEndOf="parent"
48             app:layout_constraintStart_toEndOf="@id/six"
49             app:layout_constraintTop_toTopOf="parent"
50             tools:ignore="HardcodedText" />
51
52         <View
53             android:layout_width="1dp"
54             android:layout_height="match_parent"
55             android:background="?attr/border"
56             app:layout_constraintTop_toTopOf="parent"
57             app:layout_constraintBottom_toBottomOf="parent"
58             app:layout_constraintStart_toEndOf="@id/six"
59             app:layout_constraintEnd_toStartOf="@id/multiply"/>

```

```

20
21 </androidx.constraintlayout.widget.ConstraintLayout>
22
23 <androidx.constraintlayout.widget.ConstraintLayout
24     android:id="@+id/fourthRow"
25     android:layout_width="match_parent"
26     android:layout_height="0dp"
27     app:layout_constraintBottom_toTopOf="@id/fifthRow"
28     app:layout_constraintEnd_toEndOf="parent"
29     app:layout_constraintStart_toStartOf="parent"
30     app:layout_constraintTop_toBottomOf="@id/thirdRow">
31
32     <com.google.android.material.button.MaterialButton
33         android:id="@+id/one"
34         style="@style/Widget.CalculatorPlus.NumPad.Primary"
35         android:layout_width="0dp"
36         android:layout_height="match_parent"
37         android:text="1"
38         android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
39         app:layout_constraintEnd_toStartOf="@id/two"
40         app:layout_constraintStart_toStartOf="parent"
41         app:layout_constraintTop_toTopOf="parent"
42         tools:ignore="HardcodedText" />
43
44     <com.google.android.material.button.MaterialButton
45         android:id="@+id/two"
46         style="@style/Widget.CalculatorPlus.NumPad.Primary"
47         android:layout_width="0dp"
48         android:layout_height="match_parent"
49         android:text="2"
50         android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
51         app:layout_constraintEnd_toStartOf="@id/three"
52         app:layout_constraintStart_toEndOf="@id/one"
53         app:layout_constraintTop_toTopOf="parent"
54         tools:ignore="HardcodedText" />
55
56     <com.google.android.material.button.MaterialButton
57         android:id="@+id/three"
58         style="@style/Widget.CalculatorPlus.NumPad.Primary"
59         android:layout_width="0dp"
60         android:layout_height="match_parent"
61         android:text="3"
62         android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
63         app:layout_constraintEnd_toStartOf="@id/plus"
64         app:layout_constraintStart_toEndOf="@id/two"
65         app:layout_constraintTop_toTopOf="parent"
66         tools:ignore="HardcodedText" />

```

```

25
26         <com.google.android.material.button.MaterialButton
27             android:id="@+id/plus"
28             style="@style/Widget.CalculatorPlus.NumPad.Primary"
29             android:layout_width="0dp"
30             android:layout_height="match_parent"
31             android:text="\u002B"
32             android:textColor="?attr/operatorBtnColor"
33             android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle2"
34             app:layout_constraintEnd_toEndOf="parent"
35             app:layout_constraintStart_toEndOf="@id/three"
36             app:layout_constraintTop_toTopOf="parent"
37             tools:ignore="HardcodedText" />
38
39
40     <View
41         android:layout_width="1dp"
42         android:layout_height="match_parent"
43         android:background="?attr/border"
44         app:layout_constraintTop_toTopOf="parent"
45         app:layout_constraintBottom_toBottomOf="parent"
46         app:layout_constraintStart_toEndOf="@id/three"
47         app:layout_constraintEnd_toStartOf="@id/plus"/>
48
49 </androidx.constraintlayout.widget.ConstraintLayout>
50
51 <androidx.constraintlayout.widget.ConstraintLayout
52     android:id="@+id/fifthRow"
53     android:layout_width="match_parent"
54     android:layout_height="0dp"
55     app:layout_constraintBottom_toBottomOf="parent"
56     app:layout_constraintEnd_toEndOf="parent"
57     app:layout_constraintStart_toStartOf="parent"
58     app:layout_constraintTop_toBottomOf="@id/fourthRow">
59
60     <com.google.android.material.button.MaterialButton
61         android:id="@+id/decimal"
62         style="@style/Widget.CalculatorPlus.NumPad.Primary"
63         android:layout_width="0dp"
64         android:layout_height="match_parent"
65         android:text="."
66         android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
67         app:layout_constraintEnd_toStartOf="@id/zero"
68         app:layout_constraintStart_toStartOf="parent"
69         app:layout_constraintTop_toTopOf="parent"
70         tools:ignore="HardcodedText" />
71
72     <com.google.android.material.button.MaterialButton

```



```

30         android:id="@+id/zero"
31         style="@style/Widget.CalculatorPlus.NumPad.Primary"
32         android:layout_width="0dp"
33         android:layout_height="match_parent"
34         android:text="0"
35         android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
36         app:layout_constraintEnd_toStartOf="@id/equal"
37         app:layout_constraintStart_toEndOf="@id/decimal"
38         app:layout_constraintTop_toTopOf="parent"
39         tools:ignore="HardcodedText" />
40
41     <com.google.android.material.button.MaterialButton
42         android:id="@+id/equal"
43         style="@style/Widget.CalculatorPlus.NumPad.Primary"
44         android:layout_width="0dp"
45         android:layout_height="match_parent"
46         android:text="="
47         android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
48         app:layout_constraintEnd_toStartOf="@id/minus"
49         app:layout_constraintStart_toEndOf="@id/zero"
50         app:layout_constraintTop_toTopOf="parent"
51         tools:ignore="HardcodedText" />
52
53     <com.google.android.material.button.MaterialButton
54         android:id="@+id/minus"
55         style="@style/Widget.CalculatorPlus.NumPad.Primary"
56         android:layout_width="0dp"
57         android:layout_height="match_parent"
58         android:text="\u2212"
59         android:textColor="?attr/operatorBtnColor"
60         android:stateListAnimator="@null"
61         android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle2"
62         app:layout_constraintEnd_toEndOf="parent"
63         app:layout_constraintStart_toEndOf="@id/equal"
64         app:layout_constraintTop_toTopOf="parent"
65         tools:ignore="HardcodedText" />
66
67     <View
68         android:layout_width="1dp"
69         android:layout_height="match_parent"
70         android:background="?attr/border"
71         app:layout_constraintTop_toTopOf="parent"
72         app:layout_constraintBottom_toBottomOf="parent"
73         app:layout_constraintStart_toEndOf="@id/equal"
74         app:layout_constraintEnd_toStartOf="@id/minus"/>
75
76 </androidx.constraintlayout.widget.ConstraintLayout>

```

```
35
38 </androidx.constraintlayout.widget.ConstraintLayout>
```

### 3.activity\_history.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.coordinatorlayout.widget.CoordinatorLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     xmlns:tools="http://schemas.android.com/tools"
7     xmlns:app="http://schemas.android.com/apk/res-auto"
8     tools:context=".ui.history.HistoryActivity">
9     <com.google.android.material.appbar.AppBarLayout
10         android:id="@+id/appBar"
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:layout_gravity="top"
14         android:backgroundTint="?attr/colorSurface">
15
16         <com.google.android.material.appbar.MaterialToolbar
17             android:id="@+id/toolbar"
18             android:layout_width="match_parent"
19             android:layout_height="?attr/actionBarSize"
20             app:navigationContentDescription="Back"
21             app:navigationIcon="@drawable/arrow_left"
22             app:title="@string/title_activity_history" />
23     </com.google.android.material.appbar.AppBarLayout>
24
25     <androidx.recyclerview.widget.RecyclerView
26         android:id="@+id/rv"
27         android:layout_width="match_parent"
28         android:layout_height="match_parent"
29         android:background="?android:attr/colorBackground"
30         app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
31
32         app:layout_behavior="com.google.android.material.appbar.AppBarLayout$ScrollingViewBehavior"
33     />
34
35     <FrameLayout
36         android:id="@+id/noHistory"
37         android:layout_width="match_parent"
38         android:layout_height="match_parent"
39         android:background="?attr/colorSurface"
```

```

38         android:visibility="gone"
39
40
41         <LinearLayout
42             android:layout_width="wrap_content"
43             android:layout_height="wrap_content"
44             android:layout_gravity="center"
45             android:orientation="horizontal">
46
47             <ImageView
48                 android:layout_width="wrap_content"
49                 android:layout_height="wrap_content"
50                 android:src="@drawable/history"
51                 android:contentDescription="@string/no_history" />
52
53             <TextView
54                 android:layout_width="wrap_content"
55                 android:layout_height="wrap_content"
56                 android:text="@string/no_history"
57                 android:layout_gravity="center"
58                 android:layout_marginStart="@dimen/space_1"
59                 android:textColor="?attr/textPrimary"
60                 android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1" />
61         </LinearLayout>
62
63     </FrameLayout>
64
65 </androidx.coordinatorlayout.widget.CoordinatorLayout>

```

## 二、函数文件

### 1.MainActivity.kt

```

1  package com.gigaworks.tech.calculator.ui.main
2
3  import android.animation.*
4  import android.content.Intent
5  import android.os.Bundle
6  import android.text.Editable
7  import android.text.TextWatcher
8  import android.util.TypedValue
9  import android.view.*
10 import android.view.animation.AccelerateDecelerateInterpolator
11 import android.view.animation.AnimationUtils

```

```
12 import android.widget.Button
13 import android.widget.Toast
14 import androidx.activity.viewModels
15 import androidx.appcompat.app.AppCompatActivity
16 import com.getkeepsafe.taptargetview.TapTarget
17 import com.getkeepsafe.taptargetview.TapTargetSequence
18 import com.gigaworks.tech.calculator.R
19 import com.gigaworks.tech.calculator.databinding.ActivityMainBinding
20 import com.gigaworks.tech.calculator.domain.History
21
22 import com.gigaworks.tech.calculator.ui.base.BaseActivity
23 import com.gigaworks.tech.calculator.ui.history.HistoryActivity
24 import com.gigaworks.tech.calculator.ui.main.helper.*
25 import com.gigaworks.tech.calculator.ui.main.viewmodel.MainViewModel
26
27 import com.gigaworks.tech.calculator.ui.view.CalculatorEditText
28 import com.gigaworks.tech.calculator.util.*
29 import dagger.hilt.android.AndroidEntryPoint
30 import java.util.*
31 import kotlin.math.sqrt
32
33
34 @AndroidEntryPoint
35 class MainActivity : BaseActivity<ActivityMainBinding>() {
36
37     private val viewModel: MainViewModel by viewModels()
38     private var mCurrentAnimator: Animator? = null
39
40     override fun onCreate(savedInstanceState: Bundle?) {
41         val appPreference = AppPreference(this)
42         val accentTheme =
43             appPreference.getStringPreference(AppPreference.ACCENT_THEME,
44             AccentTheme.BLUE.name)
45         setTheme(getAccentTheme(accentTheme))
46         super.onCreate(savedInstanceState)
47         setupActionBar(binding.toolbar)
48
49         setupView()
50         setupObservers()
51         setClickListener()
52         setAppTheme()
53     }
54
55     private val buttonClick = View.OnClickListener {
56         it.isHapticFeedbackEnabled = true
57         it.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY)
```

```

58     val text = (it as Button).text.toString()
59     val expression = removeNumberSeparator(getExpression())
60     var newExpression = handleClick(expression, text, viewModel.isPrevResult)
61     viewModel.isPrevResult = false
62     if (viewModel.getNumberSeparator() != NumberSeparator.OFF) {
63         newExpression = addNumberSeparator(
64             expression = newExpression,
65             isIndian = (viewModel.getNumberSeparator() == NumberSeparator.INDIAN)
66         )
67     }
68     setExpression(newExpression)
69 }
70
71 private val textSizeChangeListener =
72     CalculatorEditText.OnTextSizeChangeListener { textView, oldSize ->
73         // Calculate the values needed to perform the scale and translation animations,
74         // maintaining the same apparent baseline for the displayed text.
75         val textScale = oldSize / textView.textSize
76         val translationX = (1.0f - textScale) *
77             (textView.width / 2.0f - textView.paddingEnd)
78         val translationY = (1.0f - textScale) *
79             (textView.height / 2.0f - textView.paddingBottom)
80         val animatorSet = AnimatorSet()
81         animatorSet.playTogether(
82             ObjectAnimator.ofFloat(textView, View.SCALE_X, textScale, 1.0f),
83             ObjectAnimator.ofFloat(textView, View.SCALE_Y, textScale, 1.0f),
84             ObjectAnimator.ofFloat(textView, View.TRANSLATION_X, translationX, 0.0f),
85             ObjectAnimator.ofFloat(textView, View.TRANSLATION_Y, translationY, 0.0f)
86         )
87         animatorSet.duration =
88             resources.getInteger(android.R.integer.config_mediumAnimTime).toLong()
89         animatorSet.interpolator = AccelerateDecelerateInterpolator()
90         animatorSet.start()
91     }
92
93 private val expressionChangeListener = object : TextWatcher {
94     override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int, after: Int) {
95     }
96
97     override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count: Int) {
98     }
99
100    override fun afterTextChanged(s: Editable?) {
101        setResult("")
102        getResultEditText().setTextColor(getResultTextColor())
103        if (!removeNumberSeparator(s.toString()).isNumber()) {
104            viewModel.calculateExpression(s.toString())

```

```

10         }
11     }
12
13     }
14
15     private fun setClickListener() {
16         //number Pad
17         with(binding.numPad) {
18             //first row
19             percent.setOnClickListener(buttonClick)
20             openBracket.setOnClickListener(buttonClick)
21             closeBracket.setOnClickListener(buttonClick)
22             //second row
23             seven.setOnClickListener(buttonClick)
24             eight.setOnClickListener(buttonClick)
25             nine.setOnClickListener(buttonClick)
26             divide.setOnClickListener(buttonClick)
27             //third row
28             four.setOnClickListener(buttonClick)
29             five.setOnClickListener(buttonClick)
30             six.setOnClickListener(buttonClick)
31             multiply.setOnClickListener(buttonClick)
32             //fourth row
33             one.setOnClickListener(buttonClick)
34             two.setOnClickListener(buttonClick)
35             three.setOnClickListener(buttonClick)
36             plus.setOnClickListener(buttonClick)
37             //fifth row
38             decimal.setOnClickListener(buttonClick)
39             zero.setOnClickListener(buttonClick)
40             minus.setOnClickListener(buttonClick)
41         }
42
43         //scientific Pad
44         with(binding.scientificPad) {
45             //first row
46             sin.setOnClickListener(buttonClick)
47             cos.setOnClickListener(buttonClick)
48             tan.setOnClickListener(buttonClick)
49             //second row
50             asin.setOnClickListener(buttonClick)
51             acos.setOnClickListener(buttonClick)
52             atan.setOnClickListener(buttonClick)
53             //third row
54             exponential.setOnClickListener(buttonClick)
55             log.setOnClickListener(buttonClick)
56             naturalLog.setOnClickListener(buttonClick)

```

```

15         power.setOnClickListener(buttonClick)
16
17         //fourth row
18
19         factorial.setOnClickListener(buttonClick)
20         squareRoot.setOnClickListener(buttonClick)
21         cubeRoot.setOnClickListener(buttonClick)
22         pi.setOnClickListener(buttonClick)
23     }
24
25
26     //delete onClick
27     binding.numPad.delete.setOnClickListener {
28         it.isHapticFeedbackEnabled = true
29         it.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY)
30         val expression = removeNumberSeparator(getExpression())
31         if (expression.isEmpty()) {
32             return@setOnClickListener
33         }
34         val newExpression = if (viewModel.getNumberSeparator() != NumberSeparator.OFF) {
35             addNumberSeparator(
36                 expression = handleDelete(expression),
37                 isIndian = (viewModel.getNumberSeparator() == NumberSeparator.INDIAN)
38             )
39         } else {
40             handleDelete(expression)
41         }
42         setExpression(newExpression)
43     }
44
45
46     //delete long click
47     binding.numPad.delete.setOnLongClickListener {
48         if (getExpression().isNotEmpty()) {
49             animateClear()
50         }
51         true
52     }
53
54
55     //equal onClick
56     binding.numPad.equal.setOnClickListener {
57         it.isHapticFeedbackEnabled = true
58         it.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY)
59         val expression = removeNumberSeparator(getExpression())
60         val result = getResult()
61         if (expression.isNotEmpty()) {
62             if (result.isEmpty() || !removeNumberSeparator(result).isNumber()) {
63                 val shake = AnimationUtils.loadAnimation(this, R.anim.shake)
64                 getResultEditText().setTextColor(getResultTextColor(true))
65                 val errorStringId = viewModel.error.value ?: R.string.invalid
66                 if (errorStringId == -1) {

```

```

19         setResult("")
20     } else {
21         setResult(getString(errorStringId))
22         getResultEditText().startAnimation(shake)
23     }
24 } else {
25     val balancedExpression = viewModel.getCalculatedExpression()
26     val history = History(
27         expression = balancedExpression,
28         result = result,
29         date = System.currentTimeMillis()
30     )
31     viewModel.insertHistory(history)
32     viewModel.isPrevResult = true
33     setExpressionAfterEqual(result)
34 }
35 }
36
37 //memory store click
38 binding.scientificPad.memoryStore.setOnClickListener {
39     it.isHapticFeedbackEnabled = true
40     it.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY)
41     val result = removeNumberSeparator(getResult())
42     if (result.isNumber()) {
43         Toast.makeText(this, result, Toast.LENGTH_SHORT).show()
44         viewModel.setMemory(result)
45     }
46 }
47
48 //memory restore click
49 binding.scientificPad.memoryRestore.setOnClickListener {
50     it.isHapticFeedbackEnabled = true
51     it.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY)
52     val memory = viewModel.getMemory()
53     val expression = removeNumberSeparator(getExpression())
54     var newExpression = handleConstantClick(expression, memory,
55 5   viewModel.isPrevResult)
56
57     viewModel.isPrevResult = false
58     if (viewModel.getNumberSeparator() != NumberSeparator.OFF) {
59         newExpression = addNumberSeparator(
60             expression = newExpression,
61             isIndian = (viewModel.getNumberSeparator() == NumberSeparator.INDIAN)
62         )
63     }
64     setExpression(newExpression)
65 }

```



```

24
25         binding.scientificPad.memoryAdd.setOnClickListener {
26             it.isHapticFeedbackEnabled = true
27             it.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY)
28             val memory = viewModel.getMemory()
29             val result = removeNumberSeparator(getResult())
30             if (result.isNumber() && memory.isNumber()) {
31                 val newMemory = memory.toDouble() + result.toDouble()
32                 viewModel.setMemory(newMemory.toString())
33             }
34         }
35
36         binding.scientificPad.memorySub.setOnClickListener {
37             it.isHapticFeedbackEnabled = true
38             it.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY)
39             val memory = viewModel.getMemory()
40             val result = removeNumberSeparator(getResult())
41             if (result.isNumber() && memory.isNumber()) {
42                 val newMemory = memory.toDouble() - result.toDouble()
43                 viewModel.setMemory(newMemory.toString())
44             }
45         }
46
47         binding.calculatorPadViewPager?.addScientificPadStateChangeListener {
48             binding.scientificPad.arrow.animate().rotationBy(180F).setDuration(300).start()
49         }
50
51     }
52
53     /**
54      * Setup viewModel observers to observe the data change
55      * */
56     private fun setupObservers() {
57         viewModel.result.observe(this) {
58             setResult(it)
59         }
60     }
61
62     /**
63      * Setup the views with saved or initial values
64      * */
65     private fun setupView() {
66         viewModel.updateLaunchStatistics()
67         binding.resultPad.expression.setOnTextSizeChangeListener(textSizeChangeListener)
68         binding.resultPad.expression.addTextChangedListener(expressionChangeListener)
69     }
70

```

```

29     override fun onCreateOptionsMenu(menu: Menu?): Boolean {
30         menuInflater.inflate(R.menu.main_menu, menu)
31         return super.onCreateOptionsMenu(menu)
32     }
33
34     /**
35      * On back pressed, close the scientific pad if it is open
36      * else close the app.
37      * */
38     override fun onBackPressed() {
39         if (binding.calculatorPadViewPager?.currentItem == 0 || binding.calculatorPadViewPager
40 == null) {
41             super.onBackPressed()
42         } else {
43             binding.calculatorPadViewPager?.currentItem = 0
44         }
45     }
46
47     override fun onOptionsItemSelected(item: MenuItem): Boolean {
48         when (item.itemId) {
49
50             R.id.history -> {startActivity(Intent(this, HistoryActivity::class.java))
51             overridePendingTransition(R.anim.`in`,R.anim.out)}
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

```

33         return color
34     }
35
36     private fun animateClear() {
37         with(binding) {
38             val cx = clearView.right
39             val cy = clearView.bottom
40             val l = clearView.height
41             val b = clearView.width
42             val finalRadius = sqrt((l * l + b * b).toDouble()).toInt()
43             val anim = ViewAnimationUtils
44                 .createCircularReveal(clearView, cx, cy, 0f, finalRadius.toFloat())
45             clearView.visibility = View.VISIBLE
46             anim.duration =
1 resources.getInteger(android.R.integer.config_mediumAnimTime).toLong()
47             anim.addListener(object : AnimatorListenerAdapter() {
48                 override fun onAnimationEnd(animation: Animator) {
49                     setExpression("")
50                     setResult("")
51                     getResultEditText().setTextColor(getResultTextColor())
52                     clearView.visibility = View.INVISIBLE
53                     mCurrentAnimator = null
54                 }
55             })
56             mCurrentAnimator = anim
57             anim.start()
58         }
59     }
60
61     private fun setExpressionAfterEqual(answer: String) {
62         // Calculate the values needed to perform the scale and translation animations,
63         // accounting for how the scale will affect the final position of the text.
64         val expression = getExpressionEditText()
65         val result = getResultEditText()
66         val resultScale = expression.getVariableTextSize(answer) / result.textSize
67         val resultTranslationX = (1.0f - resultScale) * (result.width / 2.0f -
2 result.paddingEnd)
68         val resultTranslationY = (1.0f - resultScale) *
69             (result.height / 2.0f - result.paddingBottom) +
70             (expression.bottom - result.bottom) +
71             (result.paddingBottom - expression.paddingBottom)
72         val formulaTranslationY = -expression.bottom.toFloat()
73
74         // Use a value animator to fade to the final text color over the course of the
75         // animation.
76         val resultTextColor: Int = result.currentTextColor
77         val formulaTextColor: Int = expression.currentTextColor

```

```

38     val textColorAnimator =
39         ValueAnimator.ofObject(ArgbEvaluator(), resultTextColor, formulaTextColor)
40     textColorAnimator.addUpdateListener { valueAnimator ->
41         result.setTextColor(valueAnimator.animatedValue as Int)
42     }
43     val animatorSet = AnimatorSet()
44     animatorSet.playTogether(
45         textColorAnimator,
46         ObjectAnimator.ofFloat(result, View.SCALE_X, resultScale),
47         ObjectAnimator.ofFloat(result, View.SCALE_Y, resultScale),
48         ObjectAnimator.ofFloat(result, View.TRANSLATION_X, resultTranslationX),
49         ObjectAnimator.ofFloat(result, View.TRANSLATION_Y, resultTranslationY),
50         ObjectAnimator.ofFloat(expression, View.TRANSLATION_Y, formulaTranslationY)
51     )
52     animatorSet.duration =
53         resources.getInteger(android.R.integer.config_mediumAnimTime).toLong()
54     animatorSet.interpolator = AccelerateDecelerateInterpolator()
55     animatorSet.addListener(object : AnimatorListenerAdapter() {
56
57         override fun onAnimationEnd(animation: Animator) {
58             // Reset all of the values modified during the animation.
59             result.setTextColor(resultTextColor)
60             result.scaleX = 1.0f
61             result.scaleY = 1.0f
62             result.translationX = 0.0f
63             result.translationY = 0.0f
64             expression.translationY = 0.0f
65
66             // Finally update the formula to use the current result.
67             expression.setText(answer)
68             result.setText("")
69             mCurrentAnimator = null
70         }
71     })
72     mCurrentAnimator = animatorSet
73     animatorSet.start()
74 }
75
76 private fun setAppTheme() {
77     val themeMode = when (getSelectedTheme()) {
78         AppTheme.DARK -> AppCompatDelegate.MODE_NIGHT_YES
79         AppTheme.LIGHT -> AppCompatDelegate.MODE_NIGHT_NO
80         else -> AppCompatDelegate.MODE_NIGHT_FOLLOW_SYSTEM
81     }
82     AppCompatDelegate.setDefaultNightMode(themeMode)
83 }

```

```

42     private fun getSelectedTheme(): AppTheme {
43         val themeName = viewModel.getAppTheme()
44         return try {
45             AppTheme.valueOf(themeName)
46         } catch (e: IllegalArgumentException) {
47             AppTheme.SYSTEM_DEFAULT
48         }
49     }
50
51     private fun getExpressionEditText(): CalculatorEditText {
52         return binding.resultPad.expression
53     }
54
55     private fun getResultEditText(): CalculatorEditText {
56         return binding.resultPad.result
57     }
58
59     private fun setExpression(expression: String) {
60         getExpressionEditText().setText(expression)
61     }
62
63     private fun setResult(result: String) {
64         getResultEditText().setText(result)
65     }
66
67     private fun getExpression(): String {
68         return binding.resultPad.expression.text.toString().trim()
69     }
70
71     private fun getResult(): String {
72         return binding.resultPad.result.text.toString().trim()
73     }
74
75     override fun onStart() {
76         super.onStart()
77         var savedExpression = viewModel.getSavedExpression()
78         if (viewModel.getNumberSeparator() != NumberSeparator.OFF) {
79             savedExpression = addNumberSeparator(
80                 expression = savedExpression,
81                 isIndian = (viewModel.getNumberSeparator() == NumberSeparator.INDIAN)
82             )
83         }
84         setExpression(savedExpression)
85     }
86
87     override fun onStop() {
88         super.onStop()

```

```

45         mCurrentAnimator?.end()
46         val currentExpression = removeNumberSeparator(getExpression())
47         viewModel.saveExpression(currentExpression)
48     }
49
48     override fun getViewBinding(inflater: LayoutInflater) =
1   ActivityMainBinding.inflate(inflater)
48 }

```

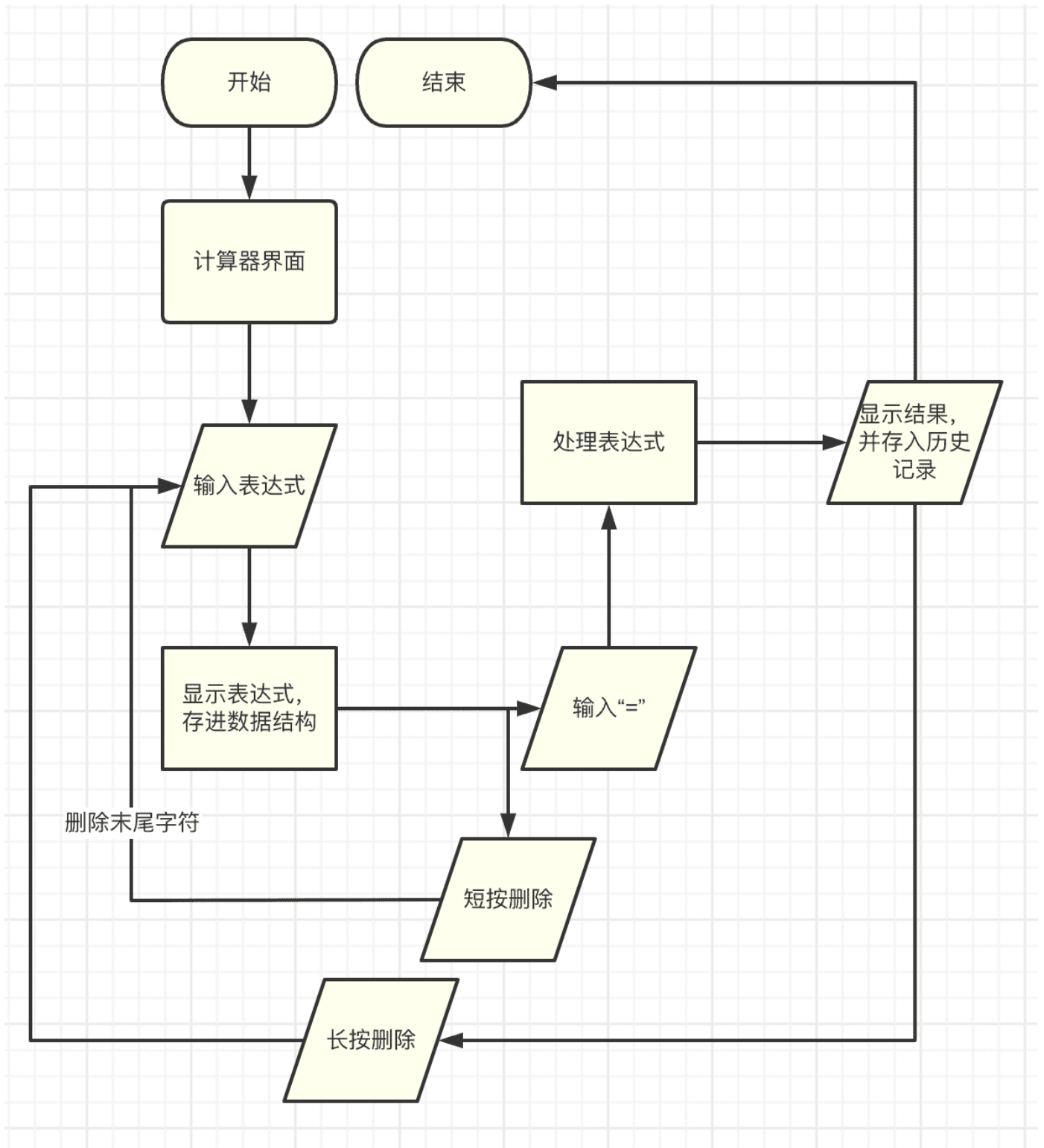
### 三、history记录

```

1
2 @Dao
3 interface HistoryDao {
4
5     @Insert(onConflict = OnConflictStrategy.REPLACE)
6     fun insertHistory(history: HistoryEntity)
7
8     @Query("SELECT * FROM history ORDER BY date DESC")
9     fun getAllHistory(): LiveData<List<HistoryEntity>>
10
11     @Query("DELETE FROM history")
12     suspend fun clearHistory()
13
14     @Query("DELETE FROM history WHERE expression=:expression")
15     suspend fun deleteHistoryByExpression(expression: String)
16
17     @Query("DELETE FROM history WHERE date < :date")
18     suspend fun deleteHistoryBefore(date: Long)
19
20 }

```

### 五、软件操作流程



## 六、难点和解决方案

1. 存储历史记录的数据，以及读取显示在activity\_history里

解决方案：用livedata存进去，然后 `getAllHistory(): LiveData<List<HistoryEntity>>` 取出来

2. 大量的按钮需要定制样式，但大部分按钮配置一样

解决方案：在styles.xml文件里定义好统一的样式

3.num\_pad 、science\_pad里按钮的位置布局

解决方案：用以下三个属性就能解决

```
1 app:layout_constraintEnd_toStartOf=""
2 app:layout_constraintStart_toEndOf=""
3 app:layout_constraintTop_toTopOf=""
```

4.实时显示表达式结果

解决方案：分为两个 `CalculatorEditText`，一个为 `expression`，一个为 `result`

## 七、不足之处

- 1.历史记录板块需要点按右上角的图标才能新建一个activity打开，应该像科学计算符号面板一样从程序顶部滑下即可
- 2.程序的动画过于简陋，应该多加点动画，使程序切换更为流畅
- 3.不能直接修改表达式中间的内容，只能从末尾一个个字符删除

## 八、今后设想

可以设计一个更好看的界面，增加自定义功能，让用户可以自由选择主色调，更好的适配深色模式。