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

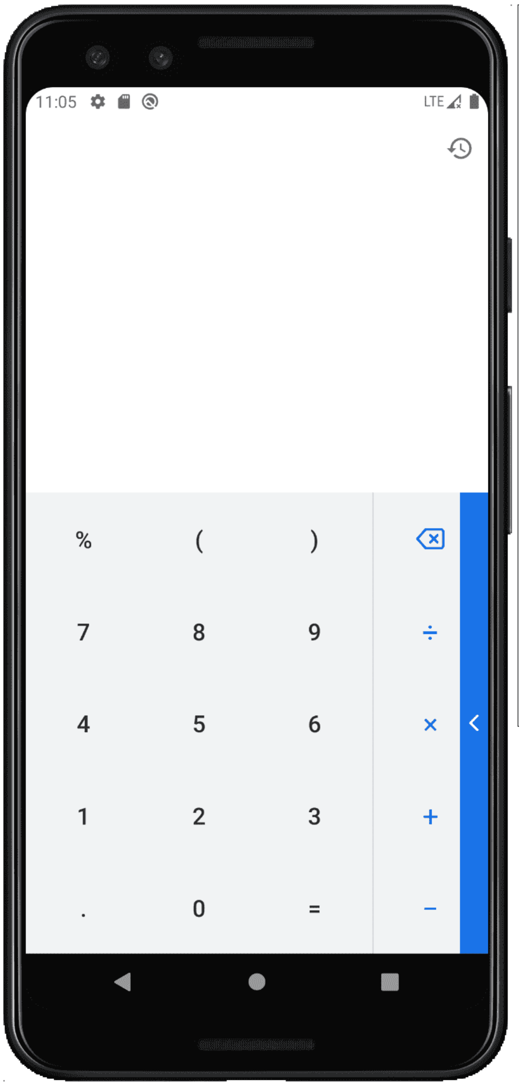
这是一个用kotlin编写的较为简易的科学计算器，软件界面参考了Android原生系统的计算器样式。可以处理常见的科学计算，有历史记录功能，可以找回曾经计算过的表达式。

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

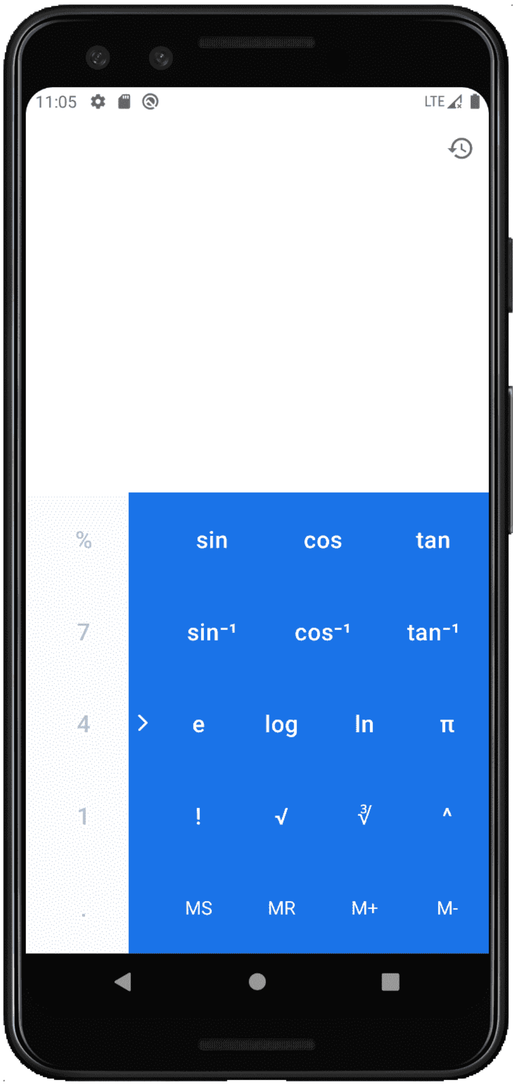
```kotlin
@Entity(tableName = "history")
data class HistoryEntity(
    @PrimaryKey(autoGenerate = false) @ColumnInfo(name = "expression") val expression: String,
    @ColumnInfo(name = "result") val result: String,
    @ColumnInfo(name = "date") val date: Long
)
```

三、界面设计

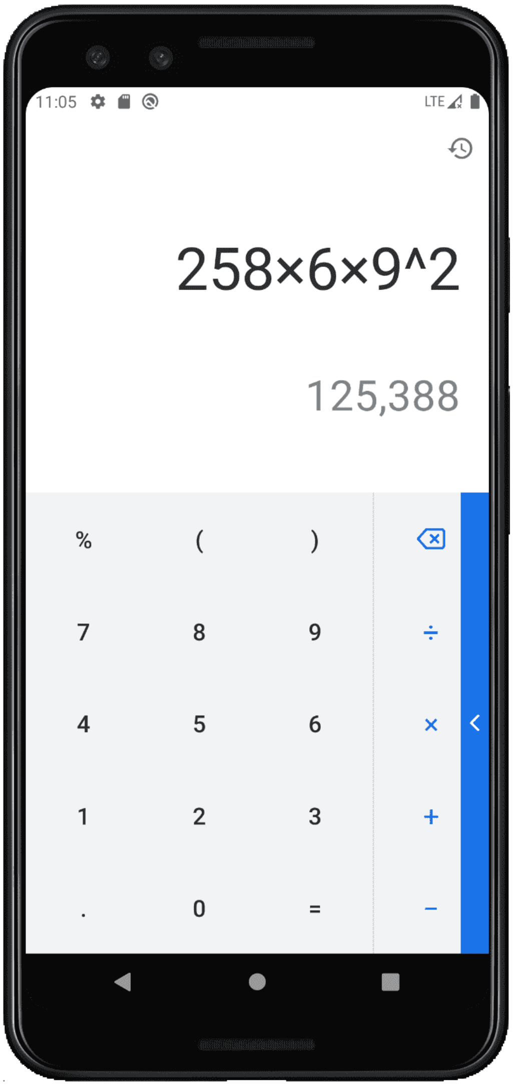
1.主界面



2.侧滑科学计算面板



3.计算结果展示

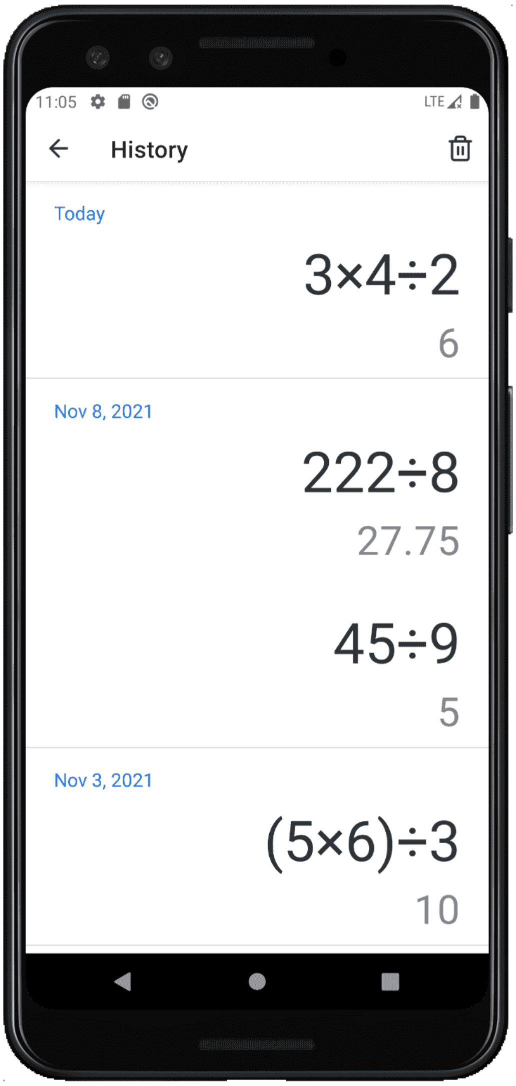


11:05

258×6×9^2

125,388

4.历史记录功能



11:05

History

Today

3×4÷2

6

Nov 8, 2021

222÷8

27.75

45÷9

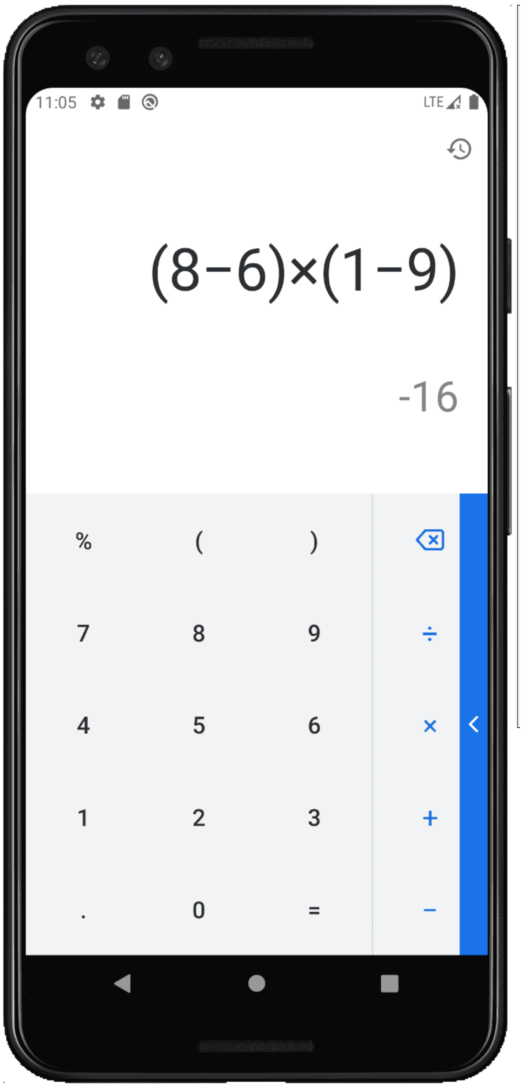
5

Nov 3, 2021

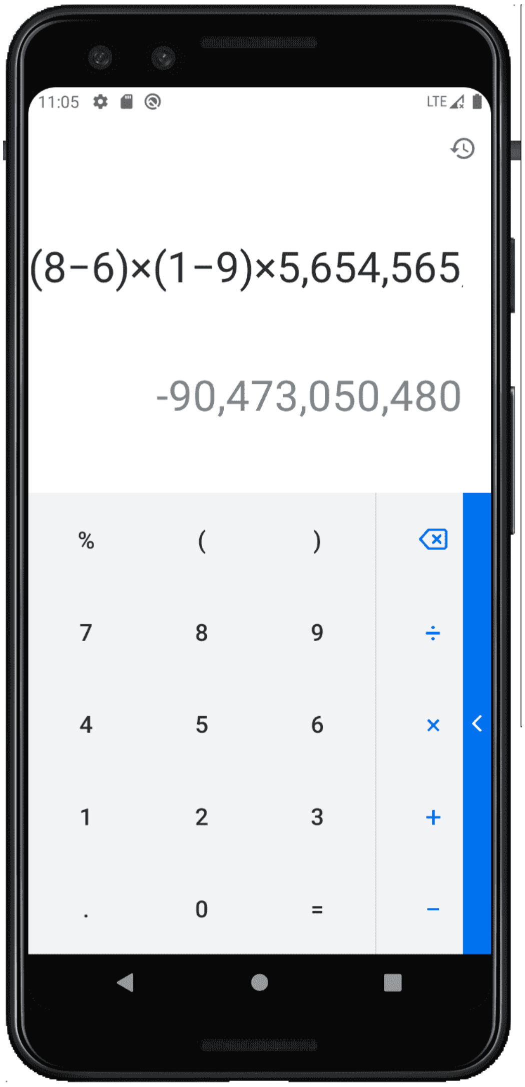
(5×6)÷3

10

5.括号运算

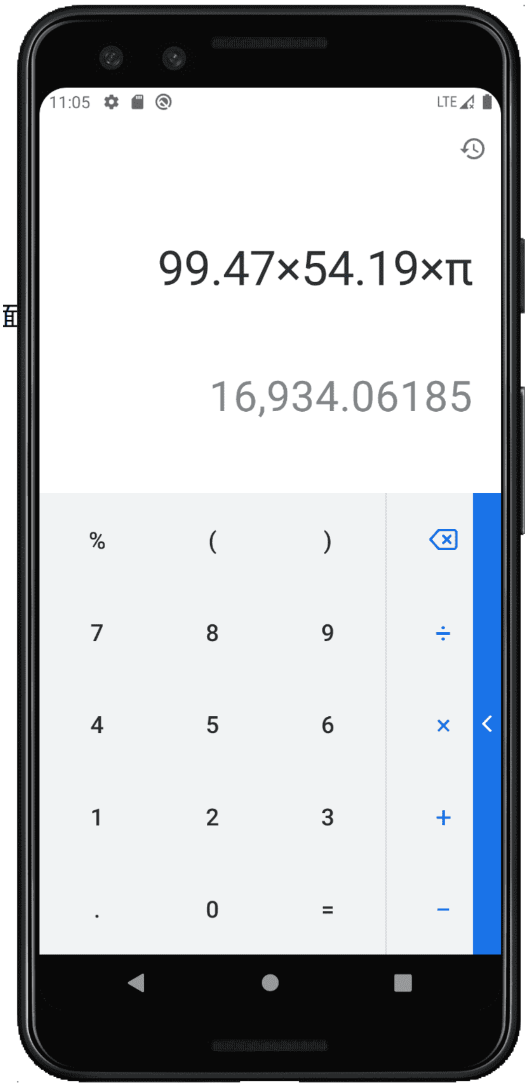


6.表达式过长自动缩小



(8−6)×(1−9)×5,654,565

-90,473,050,480



99.47×54.19×π

16,934.06185

四、关键代码

一、布局文件

1.activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ui.main.MainActivity">

    <com.google.android.material.appbar.AppBarLayout
```

```xml
        android:id="@+id/appBar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="?attr/colorSurface"
        app:elevation="0dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <com.google.android.material.appbar.MaterialToolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?actionBarSize"
            app:menu="@menu/main_menu"
            app:title=" " />
    </com.google.android.material.appbar.AppBarLayout>

    <include
        android:id="@+id/resultPad"
        layout="@layout/result_pad"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        app:layout_constraintBottom_toTopOf="@+id/guideline"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/appBar" />

    <com.gigaworks.tech.calculator.ui.view.CalculatorPadViewPager
        android:id="@+id/calculatorPadViewPager"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/guideline">

        <include
            android:id="@+id/numPad"
            layout="@layout/num_pad" />

        <include
            android:id="@+id/scientificPad"
            layout="@layout/scientific_pad" />

    </com.gigaworks.tech.calculator.ui.view.CalculatorPadViewPager>

    <androidx.constraintlayout.widget.Guideline
```

```
        android:id="@+id/guideline"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_percent="0.45" />

    <View
        android:id="@+id/clearView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:elevation="4dp"
        android:background="?attr/clearColor"
        android:visibility="invisible"
        app:layout_constraintBottom_toTopOf="@+id/guideline"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/appBar" />

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

2.num_pad.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/firstRow"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        app:layout_constraintBottom_toTopOf="@id/secondRow"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <com.google.android.material.button.MaterialButton
            android:id="@+id/percent"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text="%"
```

```xml
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
            app:layout_constraintEnd_toStartOf="@id/openBracket"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="HardcodedText" />

        <com.google.android.material.button.MaterialButton
            android:id="@+id/openBracket"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text="("
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
            app:layout_constraintEnd_toStartOf="@id/closeBracket"
            app:layout_constraintStart_toEndOf="@id/percent"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="HardcodedText" />

        <com.google.android.material.button.MaterialButton
            android:id="@+id/closeBracket"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text=")"
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
            app:layout_constraintEnd_toStartOf="@id/delete"
            app:layout_constraintStart_toEndOf="@id/openBracket"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="HardcodedText" />

        <ImageButton
            android:id="@+id/delete"
            style="@style/Widget.CalculatorPlus.NumPad.Primary.IconOnly"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:scaleType="fitCenter"
            android:src="@drawable/delete"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toEndOf="@id/closeBracket"
            app:layout_constraintTop_toTopOf="parent"
            android:contentDescription="@string/delete_button" />

        <View
            android:layout_width="1dp"
            android:layout_height="match_parent"
            android:background="?attr/border"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintStart_toEndOf="@id/closeBracket"
            app:layout_constraintEnd_toStartOf="@id/delete"/>

    </androidx.constraintlayout.widget.ConstraintLayout>

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/secondRow"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        app:layout_constraintBottom_toTopOf="@id/thirdRow"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/firstRow">

        <com.google.android.material.button.MaterialButton
            android:id="@+id/seven"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text="7"
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
            app:layout_constraintEnd_toStartOf="@id/eight"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="HardcodedText" />

        <com.google.android.material.button.MaterialButton
            android:id="@+id/eight"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text="8"
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
            app:layout_constraintEnd_toStartOf="@id/nine"
            app:layout_constraintStart_toEndOf="@id/seven"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="HardcodedText" />

        <com.google.android.material.button.MaterialButton
            android:id="@+id/nine"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text="9"
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
            app:layout_constraintEnd_toStartOf="@id/divide"
            app:layout_constraintStart_toEndOf="@id/eight"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="HardcodedText" />

        <com.google.android.material.button.MaterialButton
            android:id="@+id/divide"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text="\u00F7"
            android:textColor="?attr/operatorBtnColor"
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle2"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toEndOf="@id/nine"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="HardcodedText" />

        <View
            android:layout_width="1dp"
            android:layout_height="match_parent"
            android:background="?attr/border"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintStart_toEndOf="@id/nine"
            app:layout_constraintEnd_toStartOf="@id/divide"/>

    </androidx.constraintlayout.widget.ConstraintLayout>

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/thirdRow"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        app:layout_constraintBottom_toTopOf="@id/fourthRow"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/secondRow">

        <com.google.android.material.button.MaterialButton
            android:id="@+id/four"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text="4"
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
            app:layout_constraintEnd_toStartOf="@id/five"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
```

```xml
            tools:ignore="HardcodedText" />

        <com.google.android.material.button.MaterialButton
            android:id="@+id/five"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text="5"
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
            app:layout_constraintEnd_toStartOf="@id/six"
            app:layout_constraintStart_toEndOf="@id/four"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="HardcodedText" />

        <com.google.android.material.button.MaterialButton
            android:id="@+id/six"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text="6"
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
            app:layout_constraintEnd_toStartOf="@id/multiply"
            app:layout_constraintStart_toEndOf="@id/five"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="HardcodedText" />

        <com.google.android.material.button.MaterialButton
            android:id="@+id/multiply"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text="\u00D7"
            android:textColor="?attr/operatorBtnColor"
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle2"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toEndOf="@id/six"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="HardcodedText" />

        <View
            android:layout_width="1dp"
            android:layout_height="match_parent"
            android:background="?attr/border"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintStart_toEndOf="@id/six"
            app:layout_constraintEnd_toStartOf="@id/multiply"/>
```

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

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/fourthRow"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        app:layout_constraintBottom_toTopOf="@id/fifthRow"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/thirdRow">

        <com.google.android.material.button.MaterialButton
            android:id="@+id/one"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text="1"
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
            app:layout_constraintEnd_toStartOf="@id/two"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="HardcodedText" />

        <com.google.android.material.button.MaterialButton
            android:id="@+id/two"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text="2"
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
            app:layout_constraintEnd_toStartOf="@id/three"
            app:layout_constraintStart_toEndOf="@id/one"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="HardcodedText" />

        <com.google.android.material.button.MaterialButton
            android:id="@+id/three"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text="3"
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
            app:layout_constraintEnd_toStartOf="@id/plus"
            app:layout_constraintStart_toEndOf="@id/two"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="HardcodedText" />
```

```xml

        <com.google.android.material.button.MaterialButton
            android:id="@+id/plus"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text="\u002B"
            android:textColor="?attr/operatorBtnColor"
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle2"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toEndOf="@id/three"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="HardcodedText" />

        <View
            android:layout_width="1dp"
            android:layout_height="match_parent"
            android:background="?attr/border"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintStart_toEndOf="@id/three"
            app:layout_constraintEnd_toStartOf="@id/plus"/>

    </androidx.constraintlayout.widget.ConstraintLayout>

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/fifthRow"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/fourthRow">

        <com.google.android.material.button.MaterialButton
            android:id="@+id/decimal"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text="."
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
            app:layout_constraintEnd_toStartOf="@id/zero"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="HardcodedText" />

        <com.google.android.material.button.MaterialButton
            android:id="@+id/zero"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text="0"
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
            app:layout_constraintEnd_toStartOf="@id/equal"
            app:layout_constraintStart_toEndOf="@id/decimal"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="HardcodedText" />

        <com.google.android.material.button.MaterialButton
            android:id="@+id/equal"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text="="
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1"
            app:layout_constraintEnd_toStartOf="@id/minus"
            app:layout_constraintStart_toEndOf="@id/zero"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="HardcodedText" />

        <com.google.android.material.button.MaterialButton
            android:id="@+id/minus"
            style="@style/Widget.CalculatorPlus.NumPad.Primary"
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:text="\u2212"
            android:textColor="?attr/operatorBtnColor"
            android:stateListAnimator="@null"
            android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle2"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toEndOf="@id/equal"
            app:layout_constraintTop_toTopOf="parent"
            tools:ignore="HardcodedText" />

        <View
            android:layout_width="1dp"
            android:layout_height="match_parent"
            android:background="?attr/border"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintStart_toEndOf="@id/equal"
            app:layout_constraintEnd_toStartOf="@id/minus"/>

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

```

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

3.activity_history.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".ui.history.HistoryActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:id="@+id/appBar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="top"
        android:backgroundTint="?attr/colorSurface">

        <com.google.android.material.appbar.MaterialToolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            app:navigationContentDescription="Back"
            app:navigationIcon="@drawable/arrow_left"
            app:title="@string/title_activity_history" />
    </com.google.android.material.appbar.AppBarLayout>

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rv"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="?android:attr/colorBackground"
        app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"

 app:layout_behavior="com.google.android.material.appbar.AppBarLayout$ScrollingViewBehavior"
/>

    <FrameLayout
        android:id="@+id/noHistory"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="?attr/colorSurface"
        android:visibility="gone"
        app:layout_behavior="com.google.android.material.appbar.AppBarLayout$ScrollingViewBehavior">

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:orientation="horizontal">

            <ImageView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:src="@drawable/history"
                android:contentDescription="@string/no_history" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/no_history"
                android:layout_gravity="center"
                android:layout_marginStart="@dimen/space_l"
                android:textColor="?attr/textPrimary"
                android:textAppearance="@style/TextAppearance.CalculatorPlus.Subtitle1" />
        </LinearLayout>

    </FrameLayout>

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

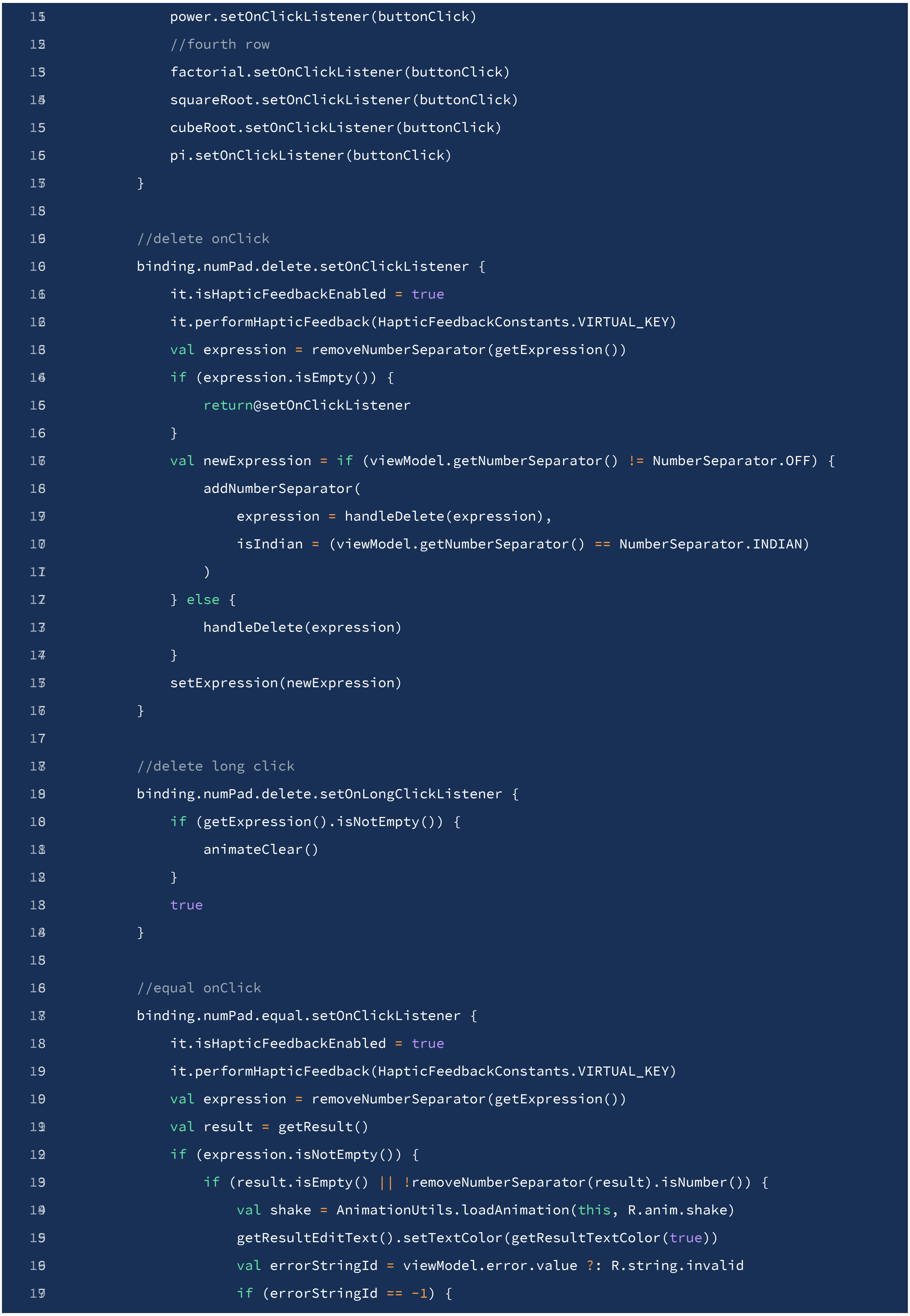
二、函数文件

1.MainActivity.kt

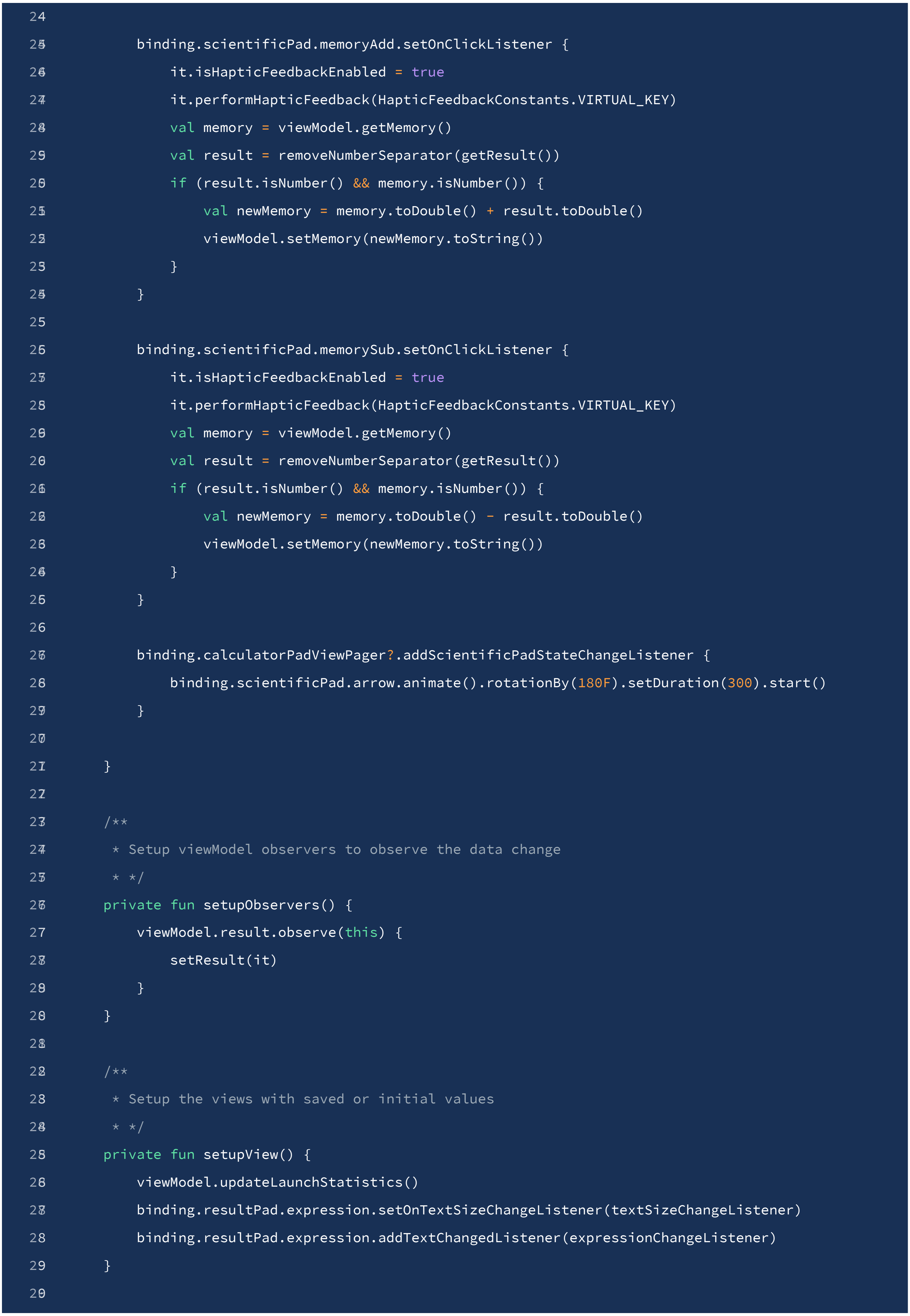
```
package com.gigaworks.tech.calculator.ui.main

import android.animation.*
import android.content.Intent
import android.os.Bundle
import android.text.Editable
import android.text.TextWatcher
import android.util.TypedValue
import android.view.*
import android.view.animation.AccelerateDecelerateInterpolator
import android.view.animation.AnimationUtils
```

```kotlin
import android.widget.Button
import android.widget.Toast
import androidx.activity.viewModels
import androidx.appcompat.app.AppCompatDelegate
import com.getkeepsafe.taptargetview.TapTarget
import com.getkeepsafe.taptargetview.TapTargetSequence
import com.gigaworks.tech.calculator.R
import com.gigaworks.tech.calculator.databinding.ActivityMainBinding
import com.gigaworks.tech.calculator.domain.History

import com.gigaworks.tech.calculator.ui.base.BaseActivity
import com.gigaworks.tech.calculator.ui.history.HistoryActivity
import com.gigaworks.tech.calculator.ui.main.helper.*
import com.gigaworks.tech.calculator.ui.main.viewmodel.MainViewModel

import com.gigaworks.tech.calculator.ui.view.CalculatorEditText
import com.gigaworks.tech.calculator.util.*
import dagger.hilt.android.AndroidEntryPoint
import java.util.*
import kotlin.math.sqrt


@AndroidEntryPoint
class MainActivity : BaseActivity<ActivityMainBinding>() {

    private val viewModel: MainViewModel by viewModels()
    private var mCurrentAnimator: Animator? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        val appPreference = AppPreference(this)
        val accentTheme =
            appPreference.getStringPreference(AppPreference.ACCENT_THEME, AccentTheme.BLUE.name)
        setTheme(getAccentTheme(accentTheme))
        super.onCreate(savedInstanceState)
        setupActionBar(binding.toolbar)

        setupView()
        setupObservers()
        setClickListener()
        setAppTheme()

    }

    private val buttonClick = View.OnClickListener {
        it.isHapticFeedbackEnabled = true
        it.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY)
```

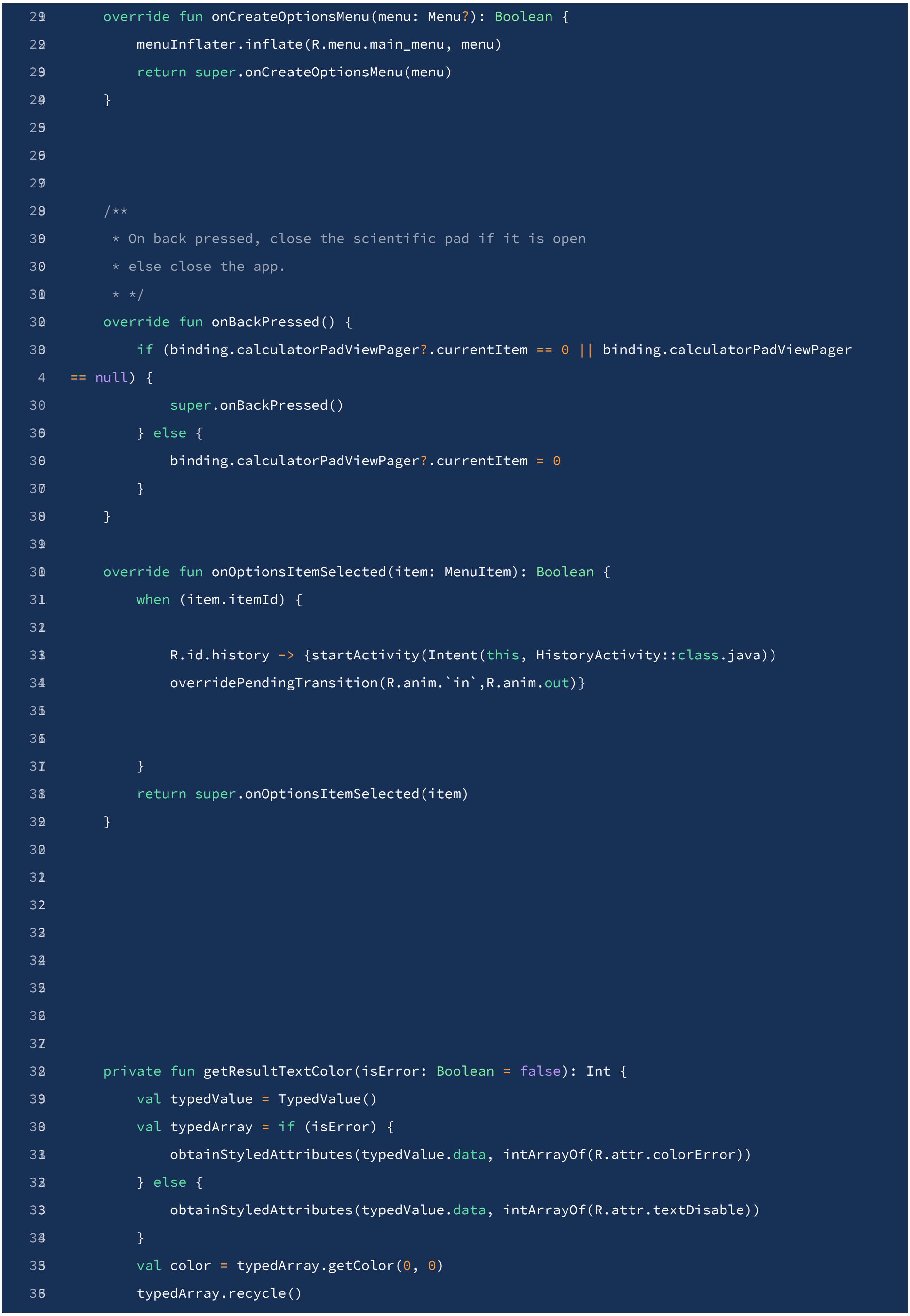
```kotlin
            val text = (it as Button).text.toString()
            val expression = removeNumberSeparator(getExpression())
            var newExpression = handleClick(expression, text, viewModel.isPrevResult)
            viewModel.isPrevResult = false
            if (viewModel.getNumberSeparator() != NumberSeparator.OFF) {
                newExpression = addNumberSeparator(
                    expression = newExpression,
                    isIndian = (viewModel.getNumberSeparator() == NumberSeparator.INDIAN)
                )
            }
            setExpression(newExpression)
        }

        private val textSizeChangeListener =
            CalculatorEditText.OnTextSizeChangeListener { textView, oldSize ->
                // Calculate the values needed to perform the scale and translation animations,
                // maintaining the same apparent baseline for the displayed text.
                val textScale = oldSize / textView.textSize
                val translationX = (1.0f - textScale) *
                        (textView.width / 2.0f - textView.paddingEnd)
                val translationY = (1.0f - textScale) *
                        (textView.height / 2.0f - textView.paddingBottom)
                val animatorSet = AnimatorSet()
                animatorSet.playTogether(
                    ObjectAnimator.ofFloat(textView, View.SCALE_X, textScale, 1.0f),
                    ObjectAnimator.ofFloat(textView, View.SCALE_Y, textScale, 1.0f),
                    ObjectAnimator.ofFloat(textView, View.TRANSLATION_X, translationX, 0.0f),
                    ObjectAnimator.ofFloat(textView, View.TRANSLATION_Y, translationY, 0.0f)
                )
                animatorSet.duration =
                    resources.getInteger(android.R.integer.config_mediumAnimTime).toLong()
                animatorSet.interpolator = AccelerateDecelerateInterpolator()
                animatorSet.start()
            }

        private val expressionChangeListener = object : TextWatcher {
            override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int, after: Int) {
            }

            override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count: Int) {
            }

            override fun afterTextChanged(s: Editable?) {
                setResult("")
                getResultEditText().setTextColor(getResultTextColor())
                if (!removeNumberSeparator(s.toString()).isNumber()) {
                    viewModel.calculateExpression(s.toString())
                }
            }

        }

        private fun setClickListener() {
            //number Pad
            with(binding.numPad) {
                //first row
                percent.setOnClickListener(buttonClick)
                openBracket.setOnClickListener(buttonClick)
                closeBracket.setOnClickListener(buttonClick)
                //second row
                seven.setOnClickListener(buttonClick)
                eight.setOnClickListener(buttonClick)
                nine.setOnClickListener(buttonClick)
                divide.setOnClickListener(buttonClick)
                //third row
                four.setOnClickListener(buttonClick)
                five.setOnClickListener(buttonClick)
                six.setOnClickListener(buttonClick)
                multiply.setOnClickListener(buttonClick)
                //fourth row
                one.setOnClickListener(buttonClick)
                two.setOnClickListener(buttonClick)
                three.setOnClickListener(buttonClick)
                plus.setOnClickListener(buttonClick)
                //fifth row
                decimal.setOnClickListener(buttonClick)
                zero.setOnClickListener(buttonClick)
                minus.setOnClickListener(buttonClick)
            }

            //scientific Pad
            with(binding.scientificPad) {
                //first row
                sin.setOnClickListener(buttonClick)
                cos.setOnClickListener(buttonClick)
                tan.setOnClickListener(buttonClick)
                //second row
                asin.setOnClickListener(buttonClick)
                acos.setOnClickListener(buttonClick)
                atan.setOnClickListener(buttonClick)
                //third row
                exponential.setOnClickListener(buttonClick)
                log.setOnClickListener(buttonClick)
                naturalLog.setOnClickListener(buttonClick)
```



```
            power.setOnClickListener(buttonClick)
            //fourth row
            factorial.setOnClickListener(buttonClick)
            squareRoot.setOnClickListener(buttonClick)
            cubeRoot.setOnClickListener(buttonClick)
            pi.setOnClickListener(buttonClick)
        }

        //delete onClick
        binding.numPad.delete.setOnClickListener {
            it.isHapticFeedbackEnabled = true
            it.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY)
            val expression = removeNumberSeparator(getExpression())
            if (expression.isEmpty()) {
                return@setOnClickListener
            }
            val newExpression = if (viewModel.getNumberSeparator() != NumberSeparator.OFF) {
                addNumberSeparator(
                    expression = handleDelete(expression),
                    isIndian = (viewModel.getNumberSeparator() == NumberSeparator.INDIAN)
                )
            } else {
                handleDelete(expression)
            }
            setExpression(newExpression)
        }

        //delete long click
        binding.numPad.delete.setOnLongClickListener {
            if (getExpression().isNotEmpty()) {
                animateClear()
            }
            true
        }

        //equal onClick
        binding.numPad.equal.setOnClickListener {
            it.isHapticFeedbackEnabled = true
            it.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY)
            val expression = removeNumberSeparator(getExpression())
            val result = getResult()
            if (expression.isNotEmpty()) {
                if (result.isEmpty() || !removeNumberSeparator(result).isNumber()) {
                    val shake = AnimationUtils.loadAnimation(this, R.anim.shake)
                    getResultEditText().setTextColor(getResultTextColor(true))
                    val errorStringId = viewModel.error.value ?: R.string.invalid
                    if (errorStringId == -1) {
                        setResult("")
                    } else {
                        setResult(getString(errorStringId))
                        getResultEditText().startAnimation(shake)
                    }
                } else {
                    val balancedExpression = viewModel.getCalculatedExpression()
                    val history = History(
                        expression = balancedExpression,
                        result = result,
                        date = System.currentTimeMillis()
                    )
                    viewModel.insertHistory(history)
                    viewModel.isPrevResult = true
                    setExpressionAfterEqual(result)
                }
            }
        }

        //memory store click
        binding.scientificPad.memoryStore.setOnClickListener {
            it.isHapticFeedbackEnabled = true
            it.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY)
            val result = removeNumberSeparator(getResult())
            if (result.isNumber()) {
                Toast.makeText(this, result, Toast.LENGTH_SHORT).show()
                viewModel.setMemory(result)
            }
        }

        //memory restore click
        binding.scientificPad.memoryRestore.setOnClickListener {
            it.isHapticFeedbackEnabled = true
            it.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY)
            val memory = viewModel.getMemory()
            val expression = removeNumberSeparator(getExpression())
            var newExpression = handleConstantClick(expression, memory, viewModel.isPrevResult)
            viewModel.isPrevResult = false
            if (viewModel.getNumberSeparator() != NumberSeparator.OFF) {
                newExpression = addNumberSeparator(
                    expression = newExpression,
                    isIndian = (viewModel.getNumberSeparator() == NumberSeparator.INDIAN)
                )
            }
            setExpression(newExpression)
        }
```



```kotlin

        binding.scientificPad.memoryAdd.setOnClickListener {
            it.isHapticFeedbackEnabled = true
            it.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY)
            val memory = viewModel.getMemory()
            val result = removeNumberSeparator(getResult())
            if (result.isNumber() && memory.isNumber()) {
                val newMemory = memory.toDouble() + result.toDouble()
                viewModel.setMemory(newMemory.toString())
            }
        }

        binding.scientificPad.memorySub.setOnClickListener {
            it.isHapticFeedbackEnabled = true
            it.performHapticFeedback(HapticFeedbackConstants.VIRTUAL_KEY)
            val memory = viewModel.getMemory()
            val result = removeNumberSeparator(getResult())
            if (result.isNumber() && memory.isNumber()) {
                val newMemory = memory.toDouble() - result.toDouble()
                viewModel.setMemory(newMemory.toString())
            }
        }

        binding.calculatorPadViewPager?.addScientificPadStateChangeListener {
            binding.scientificPad.arrow.animate().rotationBy(180F).setDuration(300).start()
        }

    }

    /**
     * Setup viewModel observers to observe the data change
     * */
    private fun setupObservers() {
        viewModel.result.observe(this) {
            setResult(it)
        }
    }

    /**
     * Setup the views with saved or initial values
     * */
    private fun setupView() {
        viewModel.updateLaunchStatistics()
        binding.resultPad.expression.setOnTextSizeChangeListener(textSizeChangeListener)
        binding.resultPad.expression.addTextChangedListener(expressionChangeListener)
    }

```



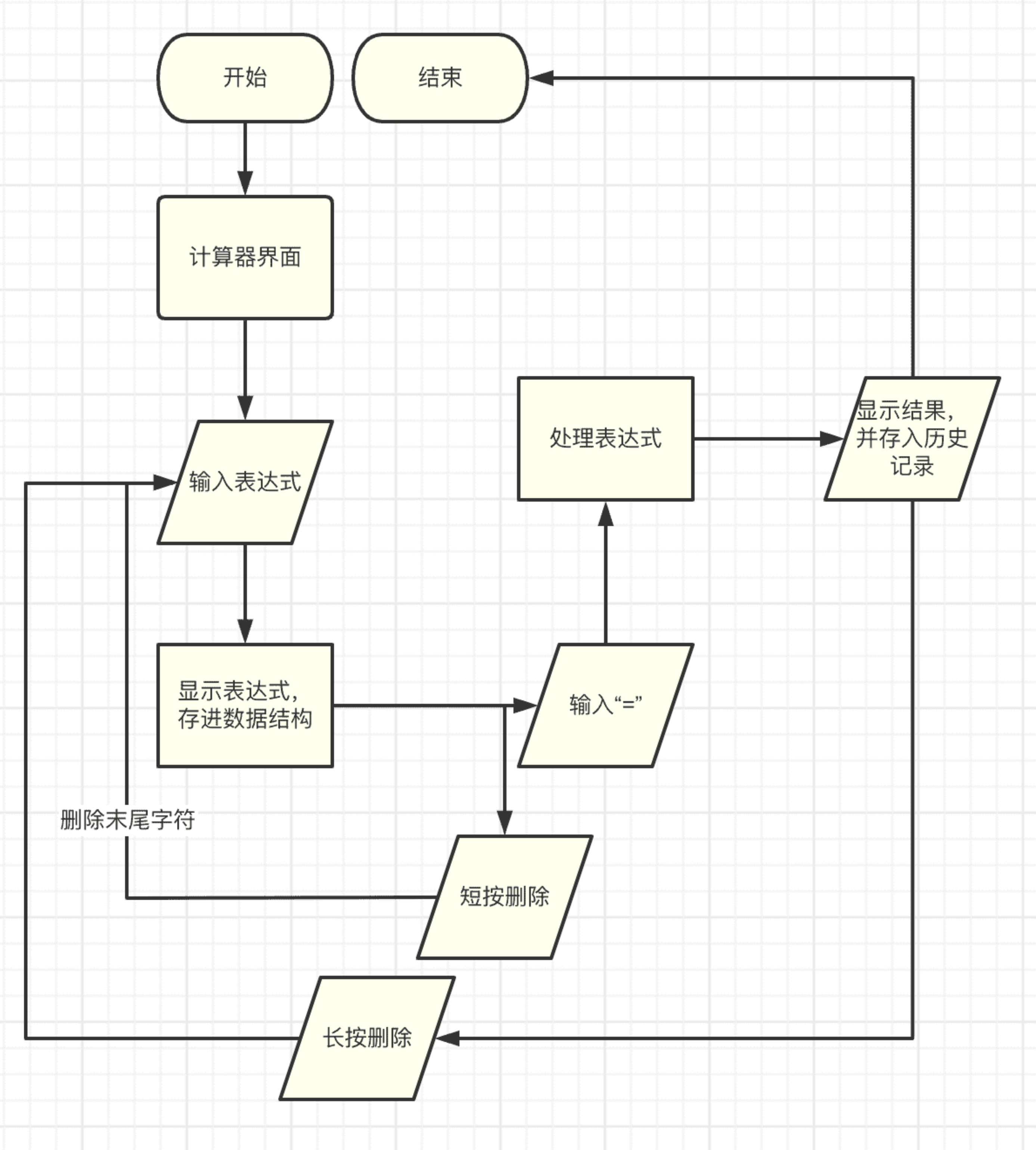
```kotlin
    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.main_menu, menu)
        return super.onCreateOptionsMenu(menu)
    }



    /**
     * On back pressed, close the scientific pad if it is open
     * else close the app.
     * */
    override fun onBackPressed() {
        if (binding.calculatorPadViewPager?.currentItem == 0 || binding.calculatorPadViewPager
== null) {
            super.onBackPressed()
        } else {
            binding.calculatorPadViewPager?.currentItem = 0
        }
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        when (item.itemId) {

            R.id.history -> {startActivity(Intent(this, HistoryActivity::class.java))
            overridePendingTransition(R.anim.`in`,R.anim.out)}


        }
        return super.onOptionsItemSelected(item)
    }







    private fun getResultTextColor(isError: Boolean = false): Int {
        val typedValue = TypedValue()
        val typedArray = if (isError) {
            obtainStyledAttributes(typedValue.data, intArrayOf(R.attr.colorError))
        } else {
            obtainStyledAttributes(typedValue.data, intArrayOf(R.attr.textDisable))
        }
        val color = typedArray.getColor(0, 0)
        typedArray.recycle()
        return color
    }

    private fun animateClear() {
        with(binding) {
            val cx = clearView.right
            val cy = clearView.bottom
            val l = clearView.height
            val b = clearView.width
            val finalRadius = sqrt((l * l + b * b).toDouble()).toInt()
            val anim = ViewAnimationUtils
                .createCircularReveal(clearView, cx, cy, 0f, finalRadius.toFloat())
            clearView.visibility = View.VISIBLE
            anim.duration =
resources.getInteger(android.R.integer.config_mediumAnimTime).toLong()
            anim.addListener(object : AnimatorListenerAdapter() {
                override fun onAnimationEnd(animation: Animator) {
                    setExpression("")
                    setResult("")
                    getResultEditText().setTextColor(getResultTextColor())
                    clearView.visibility = View.INVISIBLE
                    mCurrentAnimator = null
                }
            })
            mCurrentAnimator = anim
            anim.start()
        }
    }

    private fun setExpressionAfterEqual(answer: String) {
        // Calculate the values needed to perform the scale and translation animations,
        // accounting for how the scale will affect the final position of the text.
        val expression = getExpressionEditText()
        val result = getResultEditText()
        val resultScale = expression.getVariableTextSize(answer) / result.textSize
        val resultTranslationX = (1.0f - resultScale) * (result.width / 2.0f -
result.paddingEnd)
        val resultTranslationY = (1.0f - resultScale) *
                (result.height / 2.0f - result.paddingBottom) +
                (expression.bottom - result.bottom) +
                (result.paddingBottom - expression.paddingBottom)
        val formulaTranslationY = -expression.bottom.toFloat()

        // Use a value animator to fade to the final text color over the course of the
animation.
        val resultTextColor: Int = result.currentTextColor
        val formulaTextColor: Int = expression.currentTextColor
```

```kotlin
        val textColorAnimator =
            ValueAnimator.ofObject(ArgbEvaluator(), resultTextColor, formulaTextColor)
        textColorAnimator.addUpdateListener { valueAnimator ->
            result.setTextColor(valueAnimator.animatedValue as Int)
        }
        val animatorSet = AnimatorSet()
        animatorSet.playTogether(
            textColorAnimator,
            ObjectAnimator.ofFloat(result, View.SCALE_X, resultScale),
            ObjectAnimator.ofFloat(result, View.SCALE_Y, resultScale),
            ObjectAnimator.ofFloat(result, View.TRANSLATION_X, resultTranslationX),
            ObjectAnimator.ofFloat(result, View.TRANSLATION_Y, resultTranslationY),
            ObjectAnimator.ofFloat(expression, View.TRANSLATION_Y, formulaTranslationY)
        )
        animatorSet.duration =
            resources.getInteger(android.R.integer.config_mediumAnimTime).toLong()
        animatorSet.interpolator = AccelerateDecelerateInterpolator()
        animatorSet.addListener(object : AnimatorListenerAdapter() {

            override fun onAnimationEnd(animation: Animator) {
                // Reset all of the values modified during the animation.
                result.setTextColor(resultTextColor)
                result.scaleX = 1.0f
                result.scaleY = 1.0f
                result.translationX = 0.0f
                result.translationY = 0.0f
                expression.translationY = 0.0f

                // Finally update the formula to use the current result.
                expression.setText(answer)
                result.setText("")
                mCurrentAnimator = null
            }
        })
        mCurrentAnimator = animatorSet
        animatorSet.start()
    }

    private fun setAppTheme() {
        val themeMode = when (getSelectedTheme()) {
            AppTheme.DARK -> AppCompatDelegate.MODE_NIGHT_YES
            AppTheme.LIGHT -> AppCompatDelegate.MODE_NIGHT_NO
            else -> AppCompatDelegate.MODE_NIGHT_FOLLOW_SYSTEM
        }
        AppCompatDelegate.setDefaultNightMode(themeMode)
    }

    private fun getSelectedTheme(): AppTheme {
        val themeName = viewModel.getAppTheme()
        return try {
            AppTheme.valueOf(themeName)
        } catch (e: IllegalArgumentException) {
            AppTheme.SYSTEM_DEFAULT
        }
    }

    private fun getExpressionEditText(): CalculatorEditText {
        return binding.resultPad.expression
    }

    private fun getResultEditText(): CalculatorEditText {
        return binding.resultPad.result
    }

    private fun setExpression(expression: String) {
        getExpressionEditText().setText(expression)
    }

    private fun setResult(result: String) {
        getResultEditText().setText(result)
    }

    private fun getExpression(): String {
        return binding.resultPad.expression.text.toString().trim()
    }

    private fun getResult(): String {
        return binding.resultPad.result.text.toString().trim()
    }

    override fun onStart() {
        super.onStart()
        var savedExpression = viewModel.getSavedExpression()
        if (viewModel.getNumberSeparator() != NumberSeparator.OFF) {
            savedExpression = addNumberSeparator(
                expression = savedExpression,
                isIndian = (viewModel.getNumberSeparator() == NumberSeparator.INDIAN)
            )
        }
        setExpression(savedExpression)
    }

    override fun onStop() {
        super.onStop()
        mCurrentAnimator?.end()
        val currentExpression = removeNumberSeparator(getExpression())
        viewModel.saveExpression(currentExpression)
    }

    override fun getViewBinding(inflater: LayoutInflater) =
ActivityMainBinding.inflate(inflater)
}
```

三、history记录

```

@Dao
interface HistoryDao {

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    fun insertHistory(history: HistoryEntity)

    @Query("SELECT * FROM history ORDER BY date DESC")
    fun getAllHistory(): LiveData<List<HistoryEntity>>

    @Query("DELETE FROM history")
    suspend fun clearHistory()

    @Query("DELETE FROM history WHERE expression=:expression")
    suspend fun deleteHistoryByExpression(expression: String)

    @Query("DELETE FROM history WHERE date < :date")
    suspend fun deleteHistoryBefore(date: Long)

}
```

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

```
app:layout_constraintEnd_toStartOf=""
app:layout_constraintStart_toEndOf=""
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。