

计算器的设计与实现说明文档

作者：梁子宏 学号：20162180176

完成时间：2018.10.16

一、软件名称

美观方便的带有三角函数的计算器

二、软件内容简介

知识点使用：material design，逆波兰计算器，attr 属性和 style 样式使用，tablelayout，cardview，toolbar 的使用，activity 的跳转和转换机制。

这是一个用 Android studio 开发的一个计算器，带有三角函数的较为科学的计算器。可以先完整地写出一个等式，再对等式进行计算的一个计算器。

这款计算器的一个特色就是 拥有自定义的背景选择 和 夜间模式日间模式之间的选择。 这个功能的设计需要了解 activity 的产生机制和 activity 转换之间的状态转移。 并且需要了解 style 和 attr 属性的熟练使用。

在计算器的 model 设计上，采用了逆波兰计算器的思想，对各个运算符号进行优先级的比较，并且对 sin，cos，tan 进行字符串的替换，便于进行运算。

在计算器的布局方面， 我选择了 material design 的 toolbar，

cardview 来对计算器分割成 show_view, panel1,panel2 三个部分，然后对各个部分进行设置。

当然，这款计算器的功能并不十分完整，因为布局方面的局限，设置太多功能键似乎利用率不高，并且导致空间过于拥挤。所以只增加了括号表达式和三角函数表达式的功能。

三、界面设计

1、主界面



2、设置页界面



3、绿色界面





4、红色界面



5、夜间模式界面



6、计算功能





四、代码设计

主界面设计 cal_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="?attr/colorPrimaryDark">

    <!--toolbar 工具栏-->
    <android.support.v7.widget.Toolbar
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimaryDark"
        android:title="Caculator"
        app:title="Caculator"
        android:id="@+id/toolbar">
    </android.support.v7.widget.Toolbar>

    <!--显示界面-->
    <android.support.v7.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:background="?attr/colorPrimary">

            <TextView
                android:id="@+id/show_view_first"
                android:layout_width="match_parent"
                android:layout_height="80dp"
                android:background="?attr/colorPrimary"
                android:gravity="right|center_vertical"
                android:paddingRight="30dp"
                android:textSize="60sp" />
        </LinearLayout>
    </android.support.v7.widget.CardView>
</LinearLayout>
```

```
<TextView
    android:id="@+id/show_view_second"
    android:layout_width="match_parent"
    android:layout_height="65dp"
    android:background="?attr/colorPrimary"
    android:gravity="right"
    android:paddingRight="30dp"
    android:text="0"
    android:textSize="45sp" />

</LinearLayout>
</android.support.v7.widget.CardView>

<!-- 第一排按钮 panel1-->
<android.support.v7.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="0dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:background="?attr/panel1">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="@dimen/panel1_button_height"
            android:orientation="horizontal">

            <!-- sin-->
            <Button
                android:id="@+id/sin_button"
                android:layout_width="0dp"
                android:layout_height="match_parent"
                android:layout_weight="1"

                android:background="?attr/selectableItemBackgroundBorderless"
                android:onClick="onClick"
```

```

        android:text="sin"
        android:textAllCaps="false"

        android:textSize="@dimen/panel1_button_textsize" />

<!--cos-->
<Button
    android:id="@+id/cos_button"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"

    android:background="?attr/selectableItemBackgroundBorderless"
    android:onClick="onClick"

    android:text="cos"
    android:textAllCaps="false"
    android:textSize="@dimen/panel1_button_textsize" />

<!--tan-->
<Button
    android:id="@+id/tan_button"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"

    android:background="?attr/selectableItemBackgroundBorderless"
    android:onClick="onClick"

    android:text="tan"
    android:textAllCaps="false"
    android:textSize="@dimen/panel1_button_textsize" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="@dimen/panel1_button_height"
    android:orientation="horizontal">

    <!--左括号-->
    <Button
        android:id="@+id/leftBracket_button"

```

```

        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"

        android:background="?attr/selectableItemBackgroundBorderless"
        android:onClick="onClick"

        android:text="("
        android:textAllCaps="false"
        android:textSize="@dimen/panel1_button_textsize" />

<!--右括号-->
<Button
    android:id="@+id/rightBracket_button"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"

    android:background="?attr/selectableItemBackgroundBorderless"
    android:onClick="onClick"

    android:text=")"
    android:textAllCaps="false"
    android:textSize="@dimen/panel1_button_textsize" />

<!--删除-->
<Button
    android:id="@+id/delete_button"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"

    android:background="?attr/selectableItemBackgroundBorderless"
    android:onClick="onClick"

    android:text="del"
    android:textAllCaps="false"
    android:textSize="@dimen/panel1_button_textsize" />

</LinearLayout>

```

```

</LinearLayout>

</android.support.v7.widget.CardView>

<!--第二排按钮 panel2-->
<android.support.v7.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:layout_marginTop="0dp">

    <TableLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="?attr/panel2"
        android:orientation="vertical">

        <!--7 8 9 +-->
        <TableRow
            android:layout_height="0dp"
            android:layout_width="match_parent"
            android:layout_weight="1">

            <!--7-->
            <Button
                android:id="@+id/seven_button"
                android:layout_width="0dp"
                android:layout_height="match_parent"
                android:layout_weight="1"

                android:background="?attr/selectableItemBackgroundBorderless"
                android:onClick="onClick"

                android:text="7"
                android:textAllCaps="false"
                android:textSize="@dimen/panel2_button_textsize" />
            <!--8-->
            <Button
                android:id="@+id/eight_button"
                android:layout_width="0dp"
                android:layout_height="match_parent"
                android:layout_weight="1"

```

```

        android:background="?attr/selectableItemBackgroundBorderless"
        android:onClick="onClick"

        android:text="8"

        android:textAllCaps="false"
        android:textSize="@dimen/panel2_button_textsize" />
<!--9-->
<Button
    android:id="@+id/nine_button"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"

    android:background="?attr/selectableItemBackgroundBorderless"
    android:onClick="onClick"

    android:text="9"
    android:textAllCaps="false"
    android:textSize="@dimen/panel2_button_textsize" />
<!--+-->
<Button
    android:id="@+id/plus_button"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"

    android:background="?attr/selectableItemBackgroundBorderless"
    android:onClick="onClick"

    android:text="+"
    android:textAllCaps="false"
    android:textSize="@dimen/panel2_button_textsize" />

</TableRow>

<!--4 5 6 - -->
<TableRow
    android:layout_height="0dp"
    android:layout_width="match_parent"
    android:layout_weight="1">

```



```

<!--4-->
<Button
    android:id="@+id/four_button"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"

    android:background="?attr/selectableItemBackgroundBorderless"
    android:onClick="onClick"

    android:text="4"
    android:textAllCaps="false"
    android:textSize="@dimen/panel2_button_textsize" />
<!--5-->
<Button
    android:id="@+id/five_button"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"

    android:background="?attr/selectableItemBackgroundBorderless"
    android:onClick="onClick"

    android:text="5"
    android:textAllCaps="false"
    android:textSize="@dimen/panel2_button_textsize" />
<!--6-->
<Button
    android:id="@+id/six_button"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"

    android:background="?attr/selectableItemBackgroundBorderless"
    android:onClick="onClick"

    android:text="6"
    android:textAllCaps="false"
    android:textSize="@dimen/panel2_button_textsize" />
<!-- - -->
<Button
    android:id="@+id/minus_button"
    android:layout_width="0dp"

```

```

        android:layout_height="match_parent"
        android:layout_weight="1"

        android:background="?attr/selectableItemBackgroundBorderless"
        android:onClick="onClick"

        android:text="-"
        android:textAllCaps="false"
        android:textSize="@dimen/panel2_button_textsize" />

</TableRow>

<!--1 2 3 *-->
<TableRow
    android:layout_height="0dp"
    android:layout_width="match_parent"
    android:layout_weight="1">

    <!--1-->
    <Button
        android:id="@+id/one_button"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"

        android:background="?attr/selectableItemBackgroundBorderless"
        android:onClick="onClick"

        android:text="1"
        android:textAllCaps="false"
        android:textSize="@dimen/panel2_button_textsize" />

    <!--2-->
    <Button
        android:id="@+id/two_button"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"

        android:background="?attr/selectableItemBackgroundBorderless"
        android:onClick="onClick"

        android:text="2"

```

```

        android:textAllCaps="false"
        android:textSize="@dimen/panel2_button_textsize" />
<!--3-->
<Button
    android:id="@+id/three_button"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"

    android:background="?attr/selectableItemBackgroundBorderless"
    android:onClick="onClick"

    android:text="3"
    android:textAllCaps="false"
    android:textSize="@dimen/panel2_button_textsize" />
<!-- * -->
<Button
    android:id="@+id/multiply_button"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"

    android:background="?attr/selectableItemBackgroundBorderless"
    android:onClick="onClick"

    android:text="*"
    android:textAllCaps="false"
    android:textSize="@dimen/panel2_button_textsize" />

</TableRow>

<!-- . 0 = / -->
<TableRow
    android:layout_height="0dp"
    android:layout_width="match_parent"
    android:layout_weight="1">

    <!-- . -->
    <Button
        android:id="@+id/dot_button"
        android:layout_width="0dp"
        android:layout_height="match_parent"

```

```

        android:layout_weight="1"

        android:background="?attr/selectableItemBackgroundBorderless"
        android:onClick="onClick"

        android:text="."
        android:textAllCaps="false"
        android:textSize="@dimen/panel2_button_textsize" />
    <!--0-->

    <Button
        android:id="@+id/zero_button"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"

        android:background="?attr/selectableItemBackgroundBorderless"
        android:onClick="onClick"

        android:text="0"
        android:textAllCaps="false"
        android:textSize="@dimen/panel2_button_textsize" />
    <!--==-->

    <Button
        android:id="@+id/equal_button"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"

        android:background="?attr/selectableItemBackgroundBorderless"
        android:onClick="onClick"

        android:text="="
        android:textAllCaps="false"
        android:textSize="@dimen/panel2_button_textsize" />
    <!-- / -->

    <Button
        android:id="@+id/divide_button"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"

        android:background="?attr/selectableItemBackgroundBorderless"
        android:onClick="onClick"

```

```

        android:text="/"
        android:textAllCaps="false"
        android:textSize="@dimen/panel2_button_textsize" />

    </TableRow>

</TableLayout>

</android.support.v7.widget.CardView>

</LinearLayout>

```

设置界面设计 set_activity.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="?attr/colorPrimary"
    >

    <!--toolbar 工具栏-->
    <android.support.v7.widget.Toolbar
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimaryDark"
        app:title="setting">
    </android.support.v7.widget.Toolbar>

```

```
<!-- 选择前景色的按钮 -->
<android.support.v7.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:background="?attr/colorPrimaryDark">

    <!-- 选择前景色的线性布局 -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <!-- 变更颜色 这四个文字 -->
        <TextView
            android:id="@+id/textView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="5dp"
            android:text="变更颜色"
            android:textSize="15dp" />

        <!-- 四个按钮的线性布局 -->
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:layout_margin="5dp">

            <!-- 普通色按钮 -->
            <Button
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:layout_margin="5dp"
                android:onClick="onClick"

                android:background="#808080"
                android:id="@+id/toNormal_button"
                android:text="普通"
            />
        </LinearLayout>
    </LinearLayout>
</CardView>
```

```
<!-- 红色按钮 -->
<Button
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:layout_margin="5dp"
    android:onClick="onClick"

    android:background="@color/red"
    android:id="@+id/toRed_button"
    android:text="红色"
/>
```

```
<!-- 蓝色按钮 -->
<Button
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:layout_margin="5dp"
    android:onClick="onClick"

    android:background="@color/blue"
    android:id="@+id/toBlue_button"
    android:text="蓝色"
/>
```

```
<!-- 紫色按钮 -->
<Button
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:layout_margin="5dp"
    android:onClick="onClick"

    android:background="@color/purple"
    android:id="@+id/toPurple_button"
    android:text="紫色"
/>
```

```
<!-- 绿色按钮 -->
<Button
    android:layout_width="0dp"
```

```
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:layout_margin="5dp"
        android:onClick="onClick"

        android:background="@color/green"
        android:id="@+id/toGreen_button"
        android:text="绿色"
    />
```

```
    </LinearLayout>
</LinearLayout>
```

```
</android.support.v7.widget.CardView>
```

```
<Space
    android:layout_width="match_parent"
    android:layout_height="10dp" />
```

```
<!-- 选择 夜间模式还是正常模式 -->
```

```
<android.support.v7.widget.CardView
    android:layout_height="wrap_content"
    android:layout_width="match_parent"
    android:layout_margin="10dp"
    android:background="?attr/colorPrimaryDark"
>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="模式选择"
    android:textSize="15dp"
    android:layout_margin="5dp" />
```



```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_margin="5dp">

    <Button
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:onClick="onClick"

        android:text="日间模式"
        android:background="#ffffff"
        android:id="@+id/toLight_button"
    />

    <Button
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:onClick="onClick"

        android:text="夜间模式"
        android:background="#808080"
        android:id="@+id/toDark_button"
    />
```

```
</LinearLayout>
```

```
</LinearLayout>
```

```
</android.support.v7.widget.CardView>
```

```
</LinearLayout>
```

夜间模式，颜色转换 style.xml

```
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

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

        <item name="panel1">@color/panel1_light</item>
        <item name="panel2">@color/panel2_light</item>
    </style>

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

        <item name="panel1">@color/panel1_dark</item>
        <item name="panel2">@color/panel2_dark</item>
    </style>

    <style name="AppTheme.Light.Blue" parent="AppTheme.Light">

        <item name="android:colorPrimary">@color/blue</item>
        <item name="android:colorPrimaryDark">@color/blue_dark</item>
    </style>

    <style name="AppTheme.Dark.Blue" parent="AppTheme.Dark">

        <item name="android:colorPrimary">@color/blue</item>
        <item name="android:colorPrimaryDark">@color/blue_dark</item>
    </style>


```

```
<style name="AppTheme.Light.Cyan" parent="AppTheme.Light">
    <item name="colorPrimary">@color/cyan</item>
    <item name="colorPrimaryDark">@color/cyan_dark</item>
</style>

<style name="AppTheme.Dark.Cyan" parent="AppTheme.Dark">
    <item name="colorPrimary">@color/cyan</item>
    <item name="colorPrimaryDark">@color/cyan_dark</item>
</style>

<style name="AppTheme.Light.Gray" parent="AppTheme.Light">
    <item name="colorPrimary">@color/gray</item>
    <item name="colorPrimaryDark">@color/gray_dark</item>
</style>

<style name="AppTheme.Dark.Gray" parent="AppTheme.Dark">
    <item name="colorPrimary">@color/gray</item>
    <item name="colorPrimaryDark">@color/gray_dark</item>
</style>

<style name="AppTheme.Light.Green" parent="AppTheme.Light">
    <item name="colorPrimary">@color/green</item>
    <item name="colorPrimaryDark">@color/green_dark</item>
</style>

<style name="AppTheme.Dark.Green" parent="AppTheme.Dark">
    <item name="colorPrimary">@color/green</item>
    <item name="colorPrimaryDark">@color/green_dark</item>
</style>

<style name="AppTheme.Light.Indigo" parent="AppTheme.Light">
    <item name="colorPrimary">@color/indigo</item>
    <item name="colorPrimaryDark">@color/indigo_dark</item>
</style>

<style name="AppTheme.Dark.Indigo" parent="AppTheme.Dark">
    <item name="colorPrimary">@color/indigo</item>
    <item name="colorPrimaryDark">@color/indigo_dark</item>
</style>

<style name="AppTheme.Light.Purple" parent="AppTheme.Light">
    <item name="colorPrimary">@color/purple</item>
    <item name="colorPrimaryDark">@color/purple_dark</item>
</style>
```

```

<style name="AppTheme.Dark.Purple" parent="AppTheme.Dark">
    <item name="colorPrimary">@color/purple</item>
    <item name="colorPrimaryDark">@color/purple_dark</item>
</style>

<style name="AppTheme.Light.Red" parent="AppTheme.Light">
    <item name="colorPrimary">@color/red</item>
    <item name="colorPrimaryDark">@color/red_dark</item>
</style>

<style name="AppTheme.Dark.Red" parent="AppTheme.Dark">
    <item name="colorPrimary">@color/red</item>
    <item name="colorPrimaryDark">@color/red_dark</item>
</style>

</resources>

```

计算器 model 模块 Caculator_Model.java

```

package MODEL;

/**
 * Created by Liang Zihong on 2018/10/15.
 */

public class Calculate_Model {

    public static String calculate(String equation) {

        return RPN.getAnswer(equation);

    }
}

```

```
}
```

主界面 activity 代码

MainActivity.java

```
package com.example.liangzihong.calculator_v1;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.TextView;

import MODEL.Calculate_Model;

public class MainActivity extends AppCompatActivity {

    private Toolbar toolbar;
    private TextView upTv;
    private TextView downTv;

    private String upStr="";
    private String downStr="";
    boolean calculated=false;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        Util.onActivityCreatedTheme(this);
        super.onCreate(savedInstanceState);
        setContentView(R.layout.cal_activity);
        init();
    }

    /**
```

```

    * 用来更改夜间模式和背景颜色的
    */
@Override
protected void onStart(){
    super.onStart();
    if(Util.NeedToChange) {
        Util.changeTheme(this);
        Util.NeedToChange=false;
    }
}

//初始化
public void init(){
    toolbar=(Toolbar)findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    upTv=(TextView)findViewById(R.id.show_view_first);
    downTv=(TextView)findViewById(R.id.show_view_second);
}

//设置菜单的 setting 按钮
public boolean onCreateOptionsMenu(Menu menu){
    getMenuInflater().inflate(R.menu.toolbar,menu);
    return true;
}

//对菜单项的事件响应
public boolean onOptionsItemSelected(MenuItem item){
    switch (item.getItemId()){
        case R.id.setting:
            Intent intent=new Intent(this,SettingActivity.class);
            startActivity(intent);
            break;
    }
    return true;
}

public void onClick(View v){

```

```
switch(v.getId()){
    case R.id.one_button:
        downStr= downStr+"1";
        break;

    case R.id.two_button:
        downStr= downStr+"2";
        break;

    case R.id.three_button:
        downStr= downStr+"3";
        break;

    case R.id.four_button:
        downStr= downStr+"4";
        break;

    case R.id.five_button:
        downStr= downStr+"5";
        break;

    case R.id.six_button:
        downStr= downStr+"6";
        break;

    case R.id.seven_button:
        downStr= downStr+"7";
        break;

    case R.id.eight_button:
        downStr= downStr+"8";
        break;

    case R.id.nine_button:
        downStr= downStr+"9";
        break;

    case R.id.zero_button:
        downStr= downStr+"0";
        break;

    case R.id.sin_button:
        downStr= downStr+"sin";
```

```
        break;

    case R.id.cos_button:
        downStr= downStr+"cos";
        break;

    case R.id.tan_button:
        downStr= downStr+"tan0";
        break;

    case R.id.leftBracket_button:
        downStr= downStr+"(";
        break;

    case R.id.rightBracket_button:
        downStr= downStr+");";
        break;

    case R.id.dot_button:
        downStr= downStr+".";
        break;

    case R.id.delete_button:
        downStr=downStr.substring(0,downStr.length()-1);
        break;

    case R.id.plus_button:
        downStr= downStr+"+";
        break;

    case R.id.minus_button:
        downStr= downStr+"-";
        break;

    case R.id.multiply_button:
        downStr= downStr+"*";
        break;

    case R.id.divide_button:
        downStr= downStr+"/";
        break;

    case R.id.equal_button:
        String ans=Calculate_Model.calculate(downStr);
```



```

        upStr=downStr;
        downStr=ans+"";
        calculated=true;
        break;

    }
    upTv.setText(upStr);
    downTv.setText(downStr);

    if(calculated==true) {
        upStr = "";
        downStr = "";
        calculated=false;
    }
}
}

```

设置界面 activity 代码 settingActivity.java

```

package com.example.liangzihong.calculator_v1;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

/**
 * Created by Liang Zihong on 2018/10/15.
 */

public class SettingActivity extends AppCompatActivity {

```

```
private Button toNormal;
private Button toRed;
private Button toBlue;
private Button toGreen;
private Button toPurple;

private Button toLight;
private Button toDark;

@Override
protected void onCreate(Bundle savedInstanceState) {
    Util.onCreateTheme(this);

    super.onCreate(savedInstanceState);
    setContentView(R.layout.set_activity);

    init();
}

public void init(){

    toNormal=(Button)findViewById(R.id.toNormal_button);
    toRed=(Button)findViewById(R.id.toRed_button);
    toBlue=(Button)findViewById(R.id.toBlue_button);
    toGreen=(Button)findViewById(R.id.toGreen_button);
    toPurple=(Button)findViewById(R.id.toPurple_button);
    toLight=(Button)findViewById(R.id.toLight_button);
    toDark=(Button)findViewById(R.id.toDark_button);

}

public void onClick(View v){
    Util.secondActivity=this;
    switch (v.getId()){
        case R.id.toNormal_button:
            Util.changeColorTheme(COLOR.NORMAL);
            break;
        case R.id.toBlue_button:
            Util.changeColorTheme(COLOR.BLUE);
            break;
        case R.id.toRed_button:
            Util.changeColorTheme(COLOR.RED);
            break;
    }
}
```

```

        case R.id.toGreen_button:
            Util.changeColorTheme(COLOR.GREEN);
            break;
        case R.id.toPurple_button:
            Util.changeColorTheme(COLOR.PURPLE);
            break;
        case R.id.toLight_button:
            Util.changeLightTheme(MODE.LIGHT);
            break;
        case R.id.toDark_button:
            Util.changeLightTheme(MODE.DARK);
            break;
    }
}
}

```

负责将 activity 颜色变换的 Util 类

```

package com.example.liangzihong.calculator_v1;

import android.app.Activity;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;

/**
 * Created by Liang Zihong on 2018/10/15.
 */

enum COLOR{
    NORMAL, RED, BLUE, GREEN, PURPLE, Cyan, GRAY
}

enum MODE{
    LIGHT, DARK
}

public class Util {

```

```
public static MODE mode = MODE.LIGHT; //是否是日间模式
public static COLOR color=COLOR.GRAY;
public static boolean NeedToChange=false;
```

```
public static AppCompatActivity firstActivity;
public static AppCompatActivity secondActivity;
```

//思想是：直接改变 secondActivity 的颜色
//但是当由 setting 回到 MainActivity 时，MainActivity 会发生 onStart，此时在对 MainActivity 进行颜色改变

//改变颜色

```
public static void changeColorTheme( COLOR changeColor){
```

//只有输入颜色不一样时

```
if( color!=changeColor){
    color = changeColor;
    changeTheme(secondActivity);
    NeedToChange=true;
}
```

```
}
```

//改变日间夜间模式

```
public static void changeLightTheme(MODE changeMode){
```

```
if(mode!=changeMode){
    mode= changeMode;
    changeTheme(secondActivity);
    NeedToChange=true;
}
```

```
}
```

//被调用时，会强制关掉 第二个 activity，然后再重启

```
public static void changeTheme(Activity activity){
```

```
    activity.finish();
    activity.startActivity(new Intent(activity,activity.getClass()));
```

```
}
```

//重启前要调用的函数 或 onStart

```
public static void onActivityCreatedTheme(Activity activity)
```

```
{
```

```
switch (color)
{
    case NORMAL:
        if(mode==MODE.LIGHT){
            activity.setTheme(R.style.AppTheme_Light);
        }

        else{
            activity.setTheme(R.style.AppTheme_Dark);
        }

        break;
    case RED:
        if(mode==MODE.LIGHT){
            activity.setTheme(R.style.AppTheme_Light_Red);
        }

        else{
            activity.setTheme(R.style.AppTheme_Dark_Red);
        }
        break;

    case Cyan:
        if(mode==MODE.LIGHT){
            activity.setTheme(R.style.AppTheme_Light_Cyan);
        }

        else{
            activity.setTheme(R.style.AppTheme_Dark_Cyan);
        }
        break;

    case GREEN:
        if(mode==MODE.LIGHT){
            activity.setTheme(R.style.AppTheme_Light_Green);
        }

        else{
            activity.setTheme(R.style.AppTheme_Dark_Green);
        }
        break;

    case GRAY:
```

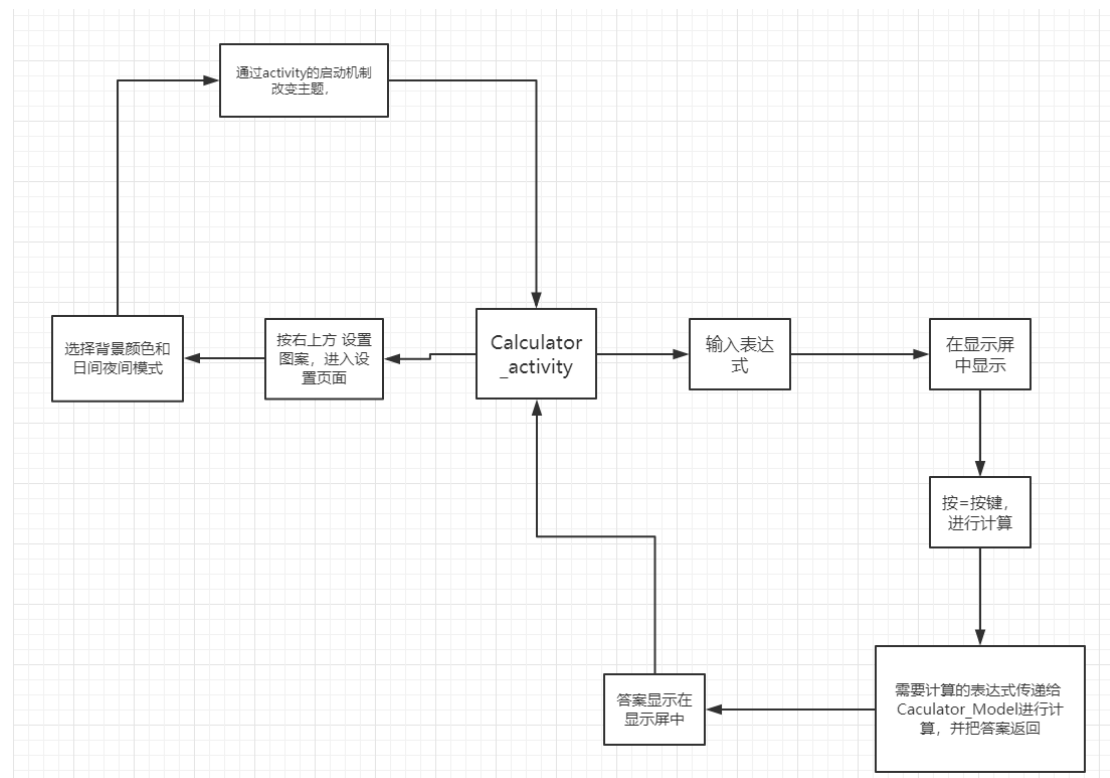
```

        if(mode==MODE.LIGHT){
            activity.setTheme(R.style.AppTheme_Light_Gray);
        }

        else{
            activity.setTheme(R.style.AppTheme_Dark_Gray);
        }
        break;
    }
    //如果是 mainActivity, 则把 Need'T'oChange 变回 false
    if(activity.getClass() == MainActivity.class && NeedToChange==true)
        NeedToChange=false;
}
}

```

五、计算器流程图



六、程序的不足之处

1、由于一开始的布局并没有考虑到更多的功能，所以只根据多数人的需求保留了 三角函数 \tan \sin \cos 函数的按键。而平方，开方取对数和阶乘并没有加入其中。

2、本来想用 `sqlite` 数据库保留每一次计算的式子，但仔细一想其实并没有这样需求，便没有设置。

3、设置页面的 `activity` 的设计过于单调，其实可以用一个 `DrawerLayout` 来代替一个 `activity`，这样可以减少 `activity` 的创建消灭时间，使内部优化更好

4、这个计算器并不能继续上一次的答案进行计算，只能每次都清零，所以这是一个之前没有想到的地方。

七、难点和解决办法

1、就对我来说，这个程序的难点就是 夜间模式和背景颜色切换的代码编写。因为所谓的 `setTheme` 函数并不能在动态中改变 `activity` 的 `theme`。只能在 `onContentView` 和 `onCreate` 前才会产生。

解决：编写一个 `Util` 类，当需要改变主题时，需要先把当前的 `activity` finish 掉，然后再重新 `startactivity`，而在 `activity` 的 `oncreate` 函数前通过 `Util` 进行 `setTheme`，这样就解决了当前页面问题。

但是对于 主 `activity`，只能重写 `activity` 的 `onStart` 函数，通过这种方式来解决。

2、 因为计算器涉及的按钮非常多，逐一将按钮 `findviewbyid` 然后逐一设置事件非常浪费时间。

解决：可以在控件的 `xml` 文档加一个 `onClick` 函数，然后在主 `activity` 中重写 `onClick` 函数即可。

八、今后对安卓开发的设想

看上去很简单的一个计算器，要想做好其实真的不简单。不仅需要编写计算表达式的 `model` 类。更要设置 计算器 UI 界面的颜色布局等等。因为老师在课堂上说过，一个移动应用最重要的就是它的 `ui` 界面，因为这是吸引用户的最重要因素。

最后，再次证实了 `cs` 的真理：纸上得来终觉浅，绝知此事要躬行。