-

華南師範大學

本科学生实验（实践）报告

院　　系：计 算 机 学 院

实验课程：移动应用开发

实验项目：计算器的设计与实现

软件名称：强力计算器

指导老师：曹阳

开课时间：2018 ～ 2019 年度第 1 学期

专　　业：计算机科学与技术

班　　级：2016 级 4 班

学　　生：余梓权

学　　号：20162180149

完成时间：2018 年 10 月 16 日

责任声明：该项目并非完全为本人原创，运算内核使用第三方开源的内核代码 Calci-kernel（https://github.com/Iraka-C/Calci-kernel），并参照项目需要进行了必要的修改；主要逻辑实现参考第三方开源项目 DarkCalculator(https://github.com/HK-SHAO/DarkCalculator)，参照《阿里巴巴 Android 开发手册》的商业化标准对整个项目进行完全重写，进行了模块分层，代码优化，UI 重定制。由于该项目还没达到真正大项目的级别，故暂时不考虑用 MVP 模式进行重构，而以常用的模块分层代之。

版本说明：截止该文档完成时间，该项目 App 为 V 1.0.0。

目录

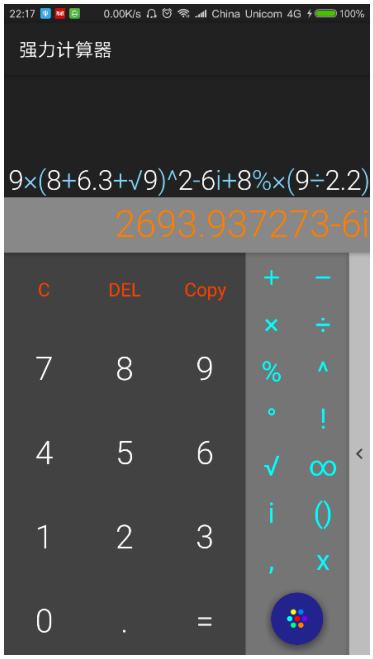
一、软件内容简介

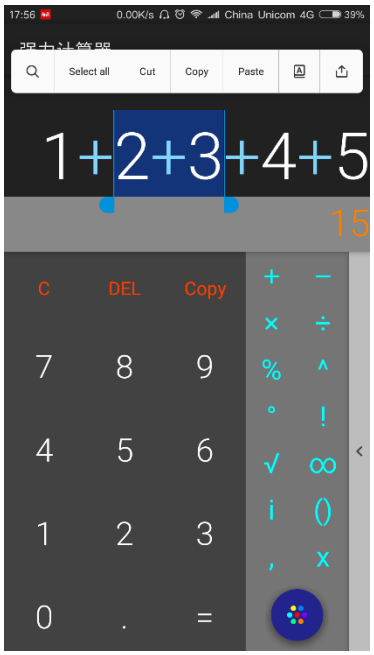
该项目实现了科学计算器的基本功能，UI 设计也追求简洁风，功能区和功能栏的设计贴近理工科的背景。支持常见的科学运算；支持常用的常数、函数运算；支持复数运算；支持大数运算；支持进制转换运算；支持数字大写转换运算；支持手机软键盘自由输入模式（上帝模式）；支持运算结果的错误提示；支持输入的表达式的各个不同部分的高亮；支持通过光标移动实时修改表达式。基本上可以满足人们日常的计算需求。

二、界面设计

主要走简洁实用的设计方向，使得用户容易上手，爱不释手。

（一）主界面

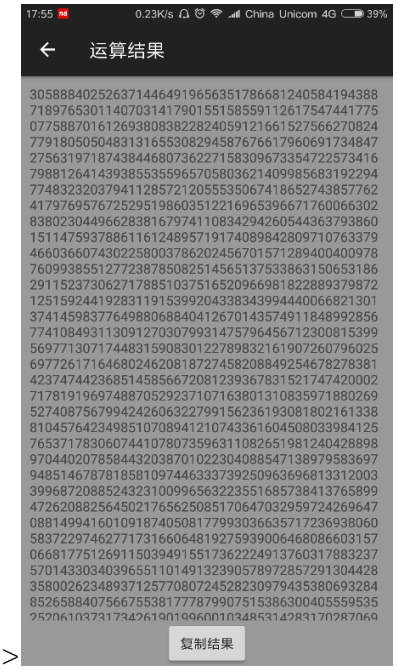


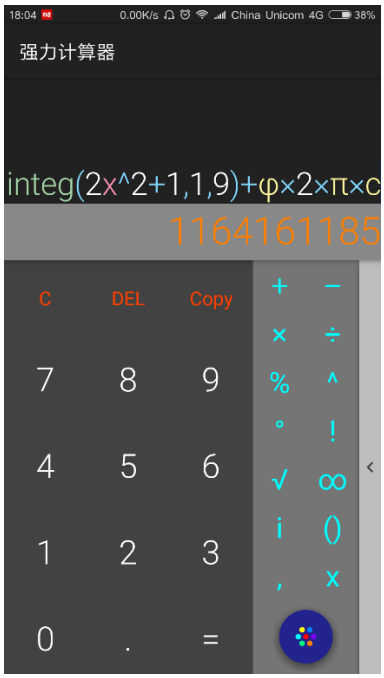

Select all
Cut
Copy
Paste
1+2+3+4+5
15
C
DEL
Copy


强力计算器
8+(5-86
括号不匹配
C
DEL
Copy


强力计算器
5555!
运算中...
C
DEL
Copy

>>>>>>


运算结果
复制结果


18:04
0.00K/s
China Unicom 4G
36%
强力计算器
integ(2x^2+1,1,9)+φ×2×π×c
1164161185
C
DEL
Copy
+
−
×
÷
7
8
9
%
^
°
!
4
5
6
√
∞
i
()
1
2
3
,
x
0
.
=

（二）功能区


22:17
0.00K/s
China Unicom 4G
100%
强力计算器
9×(8+6.3+√9)^2-6i+8%×(9÷2.2)
大数运算
进制转换
数字大写转换(财务
上帝模式
帮助
关于
退出本应用
C
DEL
Copy
+
−
÷
^
!
√
∞
i
()
x
1
2
3
,
0
.
=

1. 大数运算

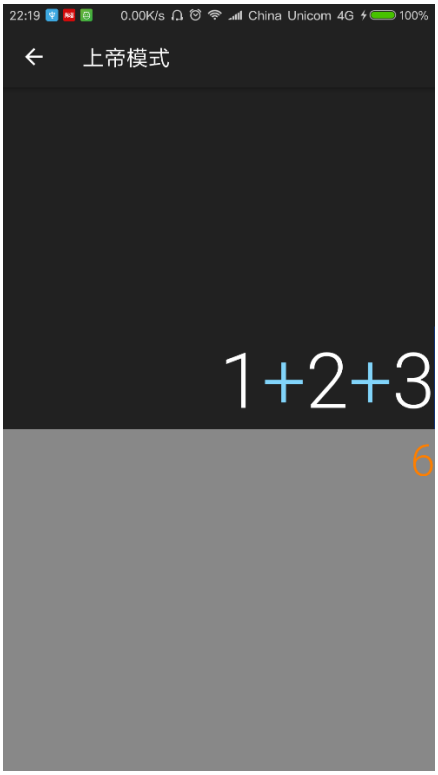

22:18 0.00K/s China Unicom 4G 100%
大数运算
操作数1：
123169191894613628261684518818585885
958589595959959565695858588565858858
285885858299565395959595695859599595
95
运算符：
+
−
×
÷
操作数2：
864346188185953658582828328284691858
258582616558265566653825466454848484
884848488588588588858858582

2. 进制转换

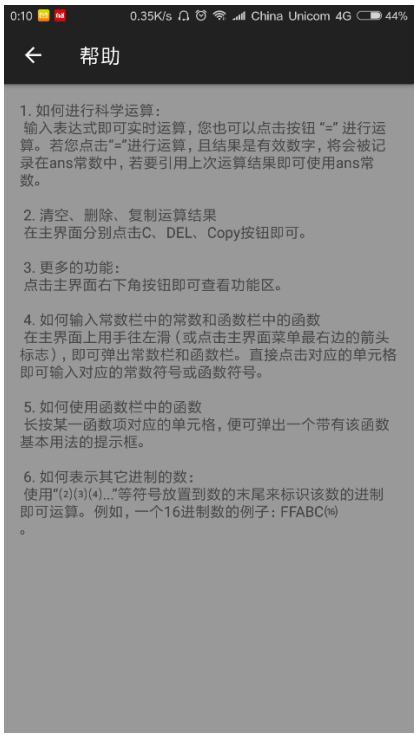

22:19 0.00K/s China Unicom 4G 100%
进制转换
待转换数字：
21318
10 进制
转换结果：
复制转换结果
1140233
5 进制

3. 数字大写转换（财务管理）

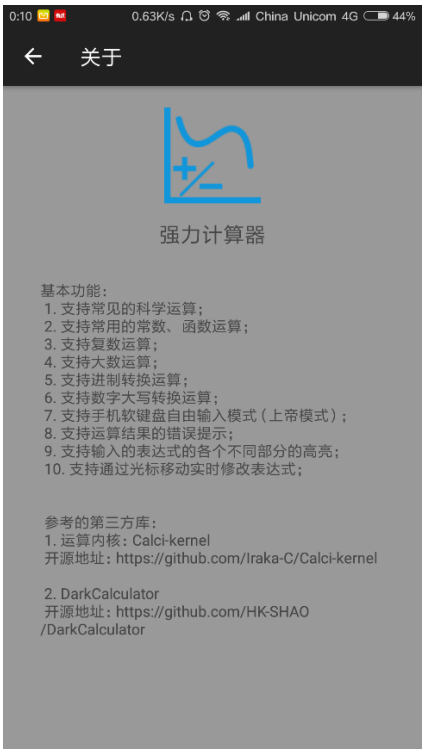
4. 上帝模式（手机软键盘自由输入模式）



5. 帮助



6. 关于



（三）功能栏

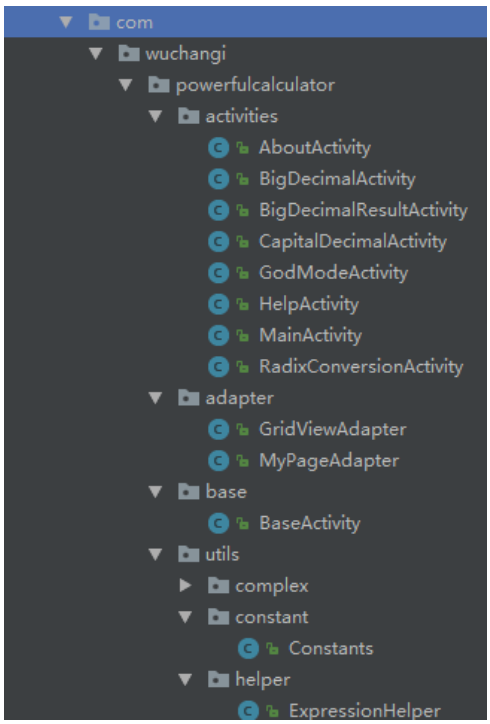
1. 常数栏

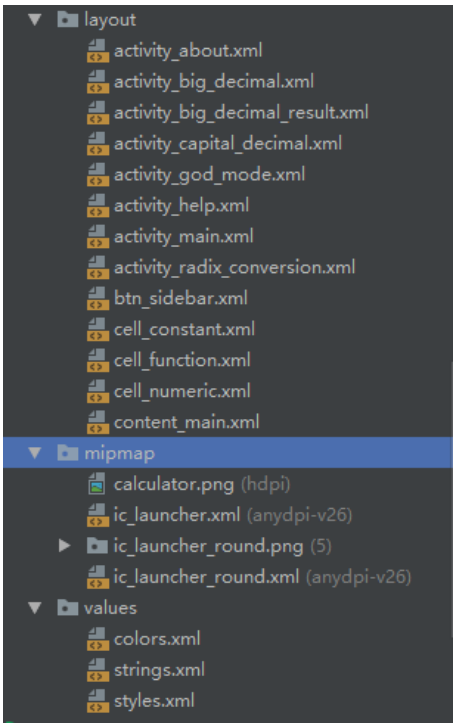
2. 函数栏

三、代码设计

鉴于本项目的代码量的缘故，这边只贴出核心部分的代码。

（一）项目分层结构

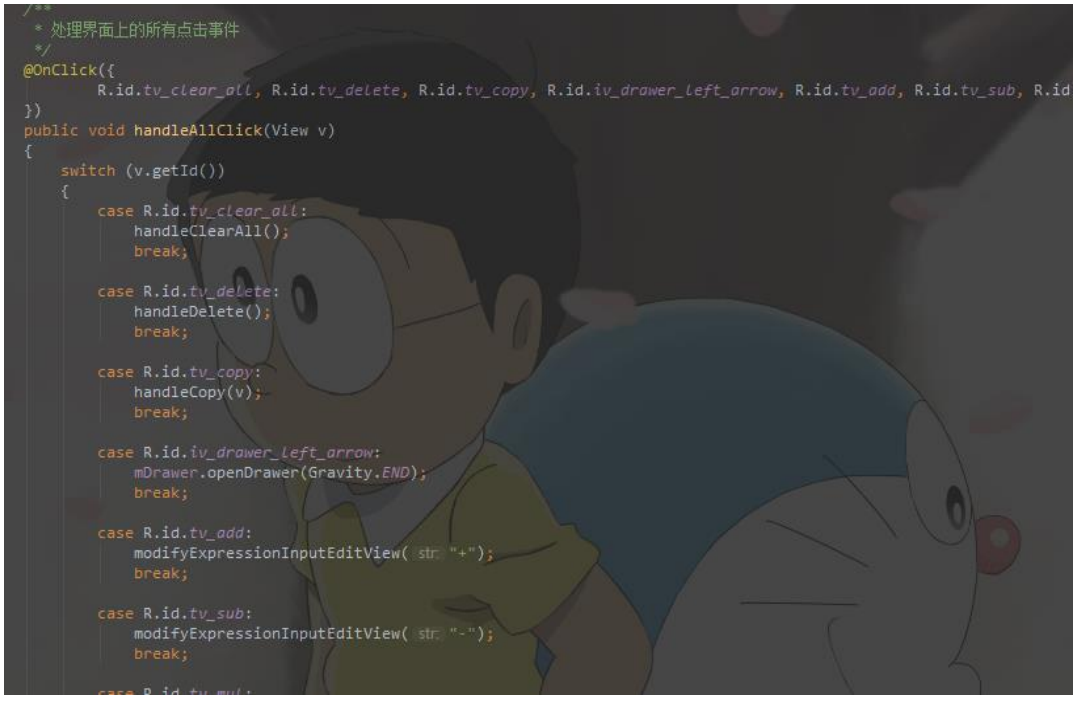




（二）MainActivity.java

```
@BindView(R.id.view_pager)
protected ViewPager mViewPager;

private BoomMenuButton mBoomMenuButton;

@BindView(R.id.et_expression_input)
protected EditText mExpressionInputEditText;

@BindView(R.id.tv_expression_result)
protected TextView mExpressionResultTextView;

/**
 * 当前是否在运算中
 */
private static boolean sIsCalculating = false;

/**
 * tab页
 */
private List<View> mPageViewList = new ArrayList<>();


@Override
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    ButterKnife.bind( target: this);

    initView();
}
```

```
/**
 * 界面初始化
 */
private void initView()
{
    initTransparentStatusBar();
    initEditText();
    initNumericMenu();
    initBMB();
    initPageViews();
    initTabPages();
}

/**...*/
private void initTransparentStatusBar()
{
    //实现透明状态栏效果
    if (Build.VERSION.SDK_INT >= 21)
    {
        View decorView = getWindow().getDecorView();

        int option = View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN | View.SYSTEM_UI_FLAG

        decorView.setSystemUiVisibility(option);

        getWindow().setStatusBarColor(Color.TRANSPARENT);
    }


    ActionBar actionBar = getSupportActionBar();
    if (actionBar != null)
    {
```



```
/**
 * 处理界面上的所有点击事件
 */
@OnClick({
        R.id.tv_clear_all, R.id.tv_delete, R.id.tv_copy, R.id.iv_drawer_left_arrow, R.id.tv_add, R.id.tv_sub, R.id
})
public void handleAllClick(View v)
{
    switch (v.getId())
    {
        case R.id.tv_clear_all:
            handleClearAll();
            break;

        case R.id.tv_delete:
            handleDelete();
            break;

        case R.id.tv_copy:
            handleCopy(v);
            break;

        case R.id.iv_drawer_left_arrow:
            mDrawer.openDrawer(Gravity.END);
            break;

        case R.id.tv_add:
            modifyExpressionInputEditView( str: "+");
            break;

        case R.id.tv_sub:
            modifyExpressionInputEditView( str: "-");
            break;
```

```
/**
 * 计算用户输入的表达式
 *
 * @param isSave 是否保存运算结果
 */
private void calcExpresssion(boolean isSave)
{
    mExpressionResultTextView.setText("运算中...");

    sIsCalculating = true;

    String expression = mExpressionInputEditText.getText().toString();

    new Thread(() ->
    {
        String[] result = ExpressionHelper.calculate(expression);

        MainActivity.this.runOnUiThread(() ->
        {
            // 运算表达式语法正确
            if (result[1].equals("false"))
            {
                if (isSave)
                {
                    // 保存运算结果
                    Constants.lastAnswerValue = result[0];
                }

                // 数值太大，跳转到大数运算结果界面查看结果
                if (result[0].getBytes().length > 1000)
                {
                    BigDecimalResultActivity.actionStart( context: MainActivity.this, result[0]);
                    return;
                }
                else
```

（三）BigDecimalActivity.java

```
            BigDecimalResultActivity.actionStart( context: this, bigDecimal1.add(bigDecimal2).toString());
            break;

        case R.id.btn_sub:
            BigDecimalResultActivity.actionStart( context: this, bigDecimal1.subtract(bigDecimal2).toString());
            break;

        case R.id.btn_mul:
            BigDecimalResultActivity.actionStart( context: this, bigDecimal1.multiply(bigDecimal2).toString());
            break;

        case R.id.btn_div:
            if(bigDecimal2.doubleValue() == 0)
            {
                Snackbar.make(v,  text: "除数不能为零", Snackbar.LENGTH_SHORT).show();
                return;
            }

            BigDecimalResultActivity.actionStart( context: this,
                    bigDecimal1.divide(bigDecimal2,  scale: 30, BigDecimal.ROUND_HALF_UP).toString());
            break;

        default:
    }
}


public static void actionStart(Context context)
{
    context.startActivity(new Intent(context, BigDecimalActivity.class));
}
```

（四）GridViewAdapter.java

```
@Override
public View getView(int position, View convertView, ViewGroup parent)
{
    ViewHolder viewHolder = null;

    if(convertView == null)
    {
        convertView = mLayoutInflater.inflate(mLayoutId, parent,  attachToRoot: false);

        viewHolder = new ViewHolder();
        viewHolder.titleTextView = convertView.findViewById(R.id.tv_title);

        if(mSubtitleList != null)
        {
            viewHolder.subTitleTextView = convertView.findViewById(R.id.tv_subtitle);
        }

        convertView.setTag(viewHolder);
    }
    else
    {
        viewHolder = (ViewHolder) convertView.getTag();
    }

    viewHolder.titleTextView.setText(mTitleList.get(position));

    if(mSubtitleList != null)
    {
        viewHolder.subTitleTextView.setText(mSubtitleList.get(position));
    }
```

（五）content_main.xml

四、软件操作流程

（一）如何进行科学运算

输入表达式即可实时运算，您也可以点击按钮 “=” 进行运算。若您点击“=”进行运算，且结果是有效数字，将会被记录在 ans 常数中，若要引用上次运算结果即可使用 ans 常数。

（二）清空、删除、复制运算结果

在主界面分别点击 C、DEL、Copy 按钮即可。

（三）更多的功能

点击主界面右下角按钮即可查看功能区。

（四）如何输入常数栏中的常数和函数栏中的函数

在主界面上用手往左滑（或点击主界面菜单最右边的箭头标志），即可弹出常数栏和函数栏。直接点击对应的单元格即可输入对应的常数符号或函数符号。

（五）如何使用函数栏中的函数

长按某一函数项对应的单元格，便可弹出一个带有该函数基本用法的提示框。

（六）如何表示其它进制的数

使用“(2)(3)(4)...”等符号放置到数的末尾来标识该数的进制即可运算。例如，一个 16 进制数的例子：FFABC(16)\n。\n

五、难点和解决方案

（一）运算内核怎么来的？

其实，以前也萌发过打造个强大的定制版计算器的想法，可以给自己日常计算使用。但是直到老师布置了该项目之后，觉得自己还是停留在用栈、中缀转后缀、C 的 atof()函数这些大二知识去处理整数、浮点数、带括号嵌套的普通表达式的阶段，顿时觉得索然无味，但是自己也不可能去打造个强大的运算内核。所以，想起了之前逛 GitHub 时发现的一位数学大神开发的一个运算内核 Calci-kernel，萌生了利用该内核的想法。摸索后发现，通过该计算内核可以非常方便地完成复杂算术表达式的运算，只需一个语法正确的算术表达式字符串，便可通过该内核使用几行代码得到结果，这就基本解决了运算内核的问题，解决了“有得用”的问题，不得不感慨拥有数学背景的程序员的强大。但是，为了定制这款计算器 App，该内核里面的某些代码表现层面和结果层面还是不能贴合项目的需要，所以又花了不少时间研究了一下内核代码，按项目需要修改了必要的代码，最终得到了定制化的内核。

（二）界面逻辑怎么实现的？

在看 Calci-kernel 的时候发现 Github 上已经有位高中的大神用 Calci-kernel 写了个强大的计算器 App（DarkCalculator），觉得它的设计和代码逻辑都很不错，代码量也很大，心中感慨万分。但是试用了一下它的这个 App 之后，觉得还是有些不足之处，比如长按切换功能的设计、侧滑菜单的设计、各个功能界面的布局（有些用户操作不方便，引导意向不明显）、代码的分层（DarkCalculator 没有分层）、代码的标准化（DarkCalculator 的代码比较乱）、代码逻辑的优化（参杂了不少多余或者不是最简的实现）、常量的管理（不少常量分散到多个 java 文件中）、没有注释说明（不方便他人阅读源代码）、一些设计细节等，掂量着目前时间还是够的，所以就打算了对该项目进行完全重写，在重写过程中也花费了大量的时间，包括了理解源代码的时间、用尽量优雅的结构结合 Android 思想重写代码、界面的重设计时间、debug 时间、测试时间等，终于完成了对 DarkCalculator 的改版升级，但是目前还没向原作者 pull request。

六、不足之处和今后的设想

（一）不足之处

没有完成对该项目的国际化（I18N），某些隐藏的 bug 还没解决。

（二）今后设想

增加一些实用的功能和附加功能，进一步优化 UI 的设计，提高用户的体验感。

七、参考文献

1、《阿里巴巴 Android 开发手册》

2、《Android 开发艺术探索》

3、https://github.com/Iraka-C/Calci-kernel

4、https://github.com/HK-SHAO/DarkCalculator