娄底职业技术学院

《基于Android开发框架基础》

实训报告

|  |
| --- |
| 题目：**Android计算器的设计与实现** |

类型：☑产品设计 □工艺设计 □方案设计

|  |  |
| --- | --- |
| 学 号： | 201702050206 |
| 班 级： | 17级软件2班 |
| 姓 名： | 周靓 |
| 指导老师： | 邓伟华 |

摘要

Android是当今最重要的手机开发平台之一，它是建立在Java基础之上的，能够迅速建立手机软件的解决方案。Android的功能十分强大，成为当今软件行业的一股新兴力量。Android基于Linux平台，由操作系统、中间件、用户界面和应用软件组成，具有以下5个特点：开放性、应用程序无界限、应用程序是在平等条件下创建的，应用程序可以轻松的嵌入网络、应用程序可以并行运行。而简单计算器又是手机上必备的小程序，所以此次创新实践很有意义。并且具有很强的使用性。

目录

[第1章 实训概述 1](#_Toc12541533)

[1.1、实训目的 1](#_Toc12541534)

[1.2、项目开发环境 1](#_Toc12541535)

[第2章、系统分析 2](#_Toc12541536)

[2.1功能分析 2](#_Toc12541537)

[2.2界面分析 2](#_Toc12541538)

[2.3进度安排 2](#_Toc12541539)

[第3章、系统设计 3](#_Toc12541540)

[3.1主界面设计 3](#_Toc12541541)

[3.2运算内容设计 8](#_Toc12541542)

[3.3活动设计 17](#_Toc12541543)

[第4章、系统测试与部署 18](#_Toc12541544)

[4.1项目开发文件夹： 18](#_Toc12541545)

[4.2系统运行界面： 18](#_Toc12541546)

[第5章、实训总结 19](#_Toc12541547)

# 第1章 实训概述

## 1.1、实训目的

通过实训设计与实现简单计算器，利用实训来培养我们运用所学课程Android的理论知识和技能，分析解决计算机实际应用中的问题的能力。

通过实训获得基本生产的感性知识，理论联系实际，扩大知识面。锻炼和培养学习业务能力及素质，接触社会、了解产业状况、培养初步担任技术工作的能力、初步了解专业实情和体验专业工作。为今后的的学习与工作，打下良好的基础。

## 1.2、项目开发环境

### 1.2.1操作系统：Windows7旗舰版

windows7旗舰版是win7所有版本中系统中功能最强大的一个版本有无限应用程序、实时[缩略图](https://baike.so.com/doc/6151218-6364416.html)预览、增强视觉体验、高级网络支持(ad-hoc无线网络和互联网连接支持[ICS](https://baike.so.com/doc/5700913-26842409.html))、移动中心(Mobility Center)。Aero Glass高级界面、高级窗口导航、改进的媒体格式支持、媒体中心和媒体流增强(包括Play To)、多点触摸、更好的手写识别等等。毛玻璃透明特效功能;多触控功能;多媒体功能(播放电影和刻录DVD);组建[家庭网络](https://baike.so.com/doc/6706653-6920651.html)组。支持加入管理网络(Domain Join)、高级[网络备份](https://baike.so.com/doc/7541759-7815852.html)和[加密文件系统](https://baike.so.com/doc/355530-376583.html)等数据保护功能、[位置感知打印](https://baike.so.com/doc/9890329-10237478.html)技术(可在家庭或办公网络上自动选择合适的[打印机](https://baike.so.com/doc/5389022-5625600.html))等等。

### 1.2.2开发工具：Android studio

Android Studio 是谷歌推出的一个Android集成开发工具，基于IntelliJ IDEA. 类似 Eclipse ADT，Android Studio 提供了集成的 Android 开发工具用于开发和调试。

### 1.2.3管理工具：git

Git是一个开源的分布式版本控制系统，可以有效、高速地处理从很小到非常大的项目版本管理。 Git 是Linus Torvalds 为了帮助管理 Linux 内核开发而开发的一个开放源码的版本控制软件。

# 第2章、系统分析

## 2.1功能分析

主要实现以下主要功能

1.计算器的主要界面

2.计算器的运算功能

## 2.2界面分析

这里创建了1个activity与MainActivity：

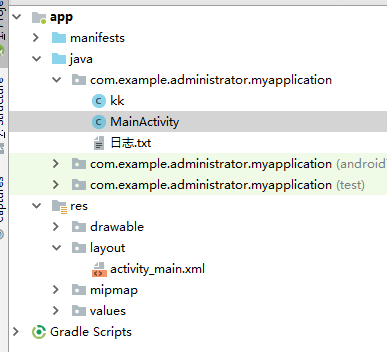


图2.1activity与MainActiyity的创建

计算器的主界面：

使用Layout在上面显示运算结果。

使用20个Button按钮分别排列显示加、减、乘、除已经归0、小数点等功能按钮。

## 2.3进度安排

6.24选择项目课题。

6.25设计项目界面。

6.26实现项目功能。

# 第3章、系统设计

## 3.1主界面设计

使用了相对布局和线性布局，最外层使用相对布局，里面套用了线性布局。

### 3.1.1界面外观

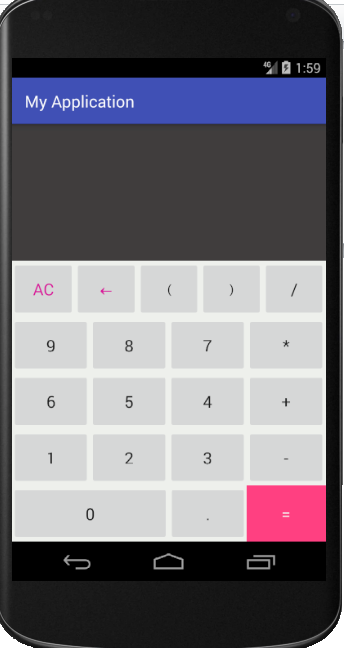


图3.1 界面外观

### 3.1.2界面关键代码：

<TextView  
 android:id="@+id/output"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="6"  
 android:gravity="end"  
 android:textColor="#faf79b"  
 android:textSize="50sp" />  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="0.6"  
 android:orientation="vertical"  
 android:showDividers="middle"  
 android:divider="#0a0a0a">  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="0.2"  
 android:orientation="horizontal"  
 android:showDividers="middle"  
 android:divider="#0a0a0a">  
 <Button  
 android:id="@+id/txtClear"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="AC"  
 android:textColor="#d8169a"  
 android:textSize="20dp" />  
 <Button  
 android:id="@+id/txtDel"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="←"  
 android:textColor="#d8169a"  
 android:textSize="20dp" />  
 <Button  
 android:id="@+id/txtLeft"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="(" />  
 <Button  
 android:id="@+id/txtRight"  
 android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text=")" />  
 <Button  
 android:id="@+id/txtDiv"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="/"  
 android:textSize="20dp" />  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="0.2"  
 android:orientation="horizontal"  
 android:showDividers="middle"  
 android:divider="#0a0a0a">  
 <Button  
 android:id="@+id/txt9"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="9"  
 android:textSize="20dp" />  
 <Button  
 android:id="@+id/txt8"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="8"  
 android:textSize="20dp" />  
 <Button  
 android:id="@+id/txt7"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="7"  
 android:textSize="20dp" />  
 <Button  
 android:id="@+id/txtMul"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="\*"  
 android:textSize="20dp" />  
 </LinearLayout>  
 <LinearLayout

android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="0.2"  
 android:orientation="horizontal"  
 android:showDividers="middle"  
 android:divider="#0a0a0a">  
 <Button  
 android:id="@+id/txt6"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="6"  
 android:textSize="20dp" />  
 <Button  
 android:id="@+id/txt5"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="5"  
 android:textSize="20dp" />  
 <Button  
 android:id="@+id/txt4"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="4"  
 android:textSize="20dp" />  
 <Button  
 android:id="@+id/txtPlus"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="+"  
 android:textSize="20dp" />  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="0.2"  
 android:orientation="horizontal"  
 android:showDividers="middle"  
 android:divider="#0a0a0a">  
 <Button  
 android:id="@+id/txt1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="1"  
 android:textSize="20dp" />  
 <Button

android:id="@+id/txt2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="2"  
 android:textSize="20dp" />  
 <Button  
 android:id="@+id/txt3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="3"  
 android:textSize="20dp" />  
 <Button  
 android:id="@+id/txtMinus"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="-"  
 android:textSize="20dp" />  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="0.2"  
 android:orientation="horizontal"  
 android:showDividers="middle"  
 android:divider="#0a0a0a">  
 <Button  
 android:id="@+id/txt0"  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="2"  
 android:text="0"  
 android:textSize="20dp" />  
 <Button  
 android:id="@+id/txtDot"  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:text="."  
 android:textSize="20sp" />  
 <Button  
 android:id="@+id/txtIs"  
 style="@android:style/Widget.DeviceDefault.Button"  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 android:layout\_weight="1"  
 android:background="@color/colorAccent"

android:text="="  
 android:textSize="20dp"></Button>  
 </LinearLayout>  
 </LinearLayout>  
</LinearLayout>

## 3.2运算内容设计

运算基本内容界面主要使用了线性布局，使用Button按钮。

### 3.2.1界面外观



图3.2 界面外观

### 3.2.2界面关键代码

**package** com.example.administrator.myapplication;  
**import** java.util.\*;  
**import** java.util.regex.Matcher;  
**import** java.util.regex.Pattern;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.TextView;  
*/\*\*  
 \** ***@author*** *yiyi  
 \* 可能有些许Bug，烦请指正。  
 \*  
 \*  
 \*/***public class** MainActivity **extends** AppCompatActivity {  
 **private int**[] **idNum** = {R.id.***txt0***, R.id.***txt1***, R.id.***txt2***, R.id.***txt3***,  
 R.id.***txt4***, R.id.***txt5***, R.id.***txt6***, R.id.***txt7***, R.id.***txt8***, R.id.***txt9***}; *//数字Number输入* **private int**[] **idCal** = {R.id.***txtPlus***, R.id.***txtMinus***, R.id.***txtMul***, R.id.***txtDiv***,R.id.***txtLeft***,R.id.***txtRight***,R.id.***txtDot***}; *//运算符* **private** Button[] **buttonsCal** = **new** Button[**idCal**.**length**];  
 **private** Button[] **buttonsNum** = **new** Button[**idNum**.**length**];  
 **private** Button **buttonEqu**; *//=* **private** Button **buttonClear**; *// AC* **private** Button **buttonDel**;  
 **private** EditText **input** ;  
 **private** TextView **output**;  
 **private static** String *Text*;  
 @Override  
 */\*\*  
 \* EditText为输入字符行，  
 \* TextView为结果。  
 \*/* **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **input** = (EditText)findViewById(R.id.***input***);  
 **input**.setText(**""**);  
 **input**.setEnabled(**false**);  
 **output** = (TextView) findViewById(R.id.***output***);

**output**.setText(**""**);  
 **buttonEqu** = (Button)findViewById(R.id.***txtIs***);  
 **buttonEqu**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **output**.setText(**new** Calculate(**input**.getText().toString()).**str**);  
 }  
 });  
 **buttonClear** = (Button) findViewById(R.id.***txtClear***);  
 **buttonClear**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **input**.setText(**""**);  
 **output**.setText(**""**);  
 }  
 });  
 **buttonDel** = (Button) findViewById(R.id.***txtDel***);  
 **buttonDel**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (!**input**.getText().toString().isEmpty() ) {  
 *Text* = **input**.getText().toString();  
 *Text* = *Text*.substring(0, *Text*.length() - 1);  
 **input**.setText(*Text*);  
 }  
 }  
 });  
*/\*\*  
 \* 注册单击事件  
 \*/* **for** (**int** idcal = 0; idcal < **idCal**.**length**; idcal++) {  
 **buttonsCal**[idcal] = (Button) findViewById(**idCal**[idcal]);  
 **buttonsCal**[idcal].setOnClickListener(**new** CalOnClick(**buttonsCal**[idcal].getText().toString()));  
 }  
 **for** (**int** i = 0; i < **idNum**.**length**; i++) {  
 **buttonsNum**[i] = (Button) findViewById(**idNum**[i]);  
 **buttonsNum**[i].setOnClickListener(**new** NumberOnClick(**buttonsNum**[i].getText().toString()));  
 }  
 }  
 *//继承OnClick接口* **class** NumberOnClick **implements** View.OnClickListener {

String **Msg**;  
 */\*\*  
 \*  
 \** ***@param msg*** *点击按钮传入字符  
 \*/* **public** NumberOnClick(String msg) {  
 **Msg** = msg;  
 }  
 @Override  
 **public void** onClick(View v) {  
 **if** (!**output**.getText().toString().equals(**""**)) {  
 **input**.setText(**""**);  
 **output**.setText(**""**);  
 }  
 **input**.append(**Msg**);  
 }  
 }  
 **class** CalOnClick **implements** View.OnClickListener{  
 String **Msg**;  
 String[] **calSymbol** = {**"+"**, **"-"**, **"\*"**, **"/"**,**"."**};  
 **public** CalOnClick(String msg) {  
 **Msg** = msg;  
 }  
 @Override  
 **public void** onClick(View v) {  
 **if** (!**output**.getText().toString().equals(**""**)) {  
 **input**.setText(**""**);  
 **output**.setText(**""**);  
 }  
 *// 检查是否运算符重复输入* **for** (**int** i = 0; i < **calSymbol**.**length**; i++) {  
 **if** (**Msg**.equals(**calSymbol**[i])) {  
 **if** (**input**.getText().toString().split(**""**)  
 [**input**.getText().toString().split(**""**).**length** - 1].equals(**calSymbol**[i])) {  
 **Msg** = **""**;  
 }  
 }  
 }  
 **input**.append(**Msg**);  
 }

}  
 */\*\*  
 \* 运算类，返回一个String结果  
 \*/* **public class** Calculate {  
 **public** String **s1**;  
 StringBuilder **str**;  
 **public** Calculate(String m) {  
 **this**.**s1** = m;  
 **try** {  
 eval();  
 } **catch** (Exception e) {  
 **str**.delete(0, **str**.length());  
 **str**.append(**"ERROR"**);  
 }  
 }  
 */\*\*  
 \*中缀表达式转后缀表达式  
 \*  
 \*遍历中缀的list  
 \*1、数字时，加入后缀list  
 \*2、“(”时，压栈  
 \*3、 若为 ')'，则依次弹栈,把弹出的运算符加入后缀表达式中，直到出现'('；  
 \*4、若为运算符，对做如下处置  
 \* 1、如果栈为空，则压栈  
 \* 2、如果栈不为空:  
 \* 1、stack.peek().equals("(") 则压栈  
 \* 2、比较str和stack.peek()的优先级  
 \* 1、如果>,则运算符压栈  
 \* 2、<=的情况：当栈不为空时:  
 \* 1、stack.peek()是左括号，压栈  
 \* 2、<=,把peek加入后缀表达式，弹栈  
 \* 3、>，把运算符压栈，停止对栈的操作  
 \* 执行完栈的操作之后，还得判断:如果栈为空,运算符压栈  
 \*/* **public** List<String> midToAfter(List<String> midList)**throws** EmptyStackException{  
 List<String> afterList=**new** ArrayList<String>();  
 Stack<String> stack=**new** Stack<String>();  
 **for**(String str:midList){  
 **int** flag=**this**.matchWitch(str);  
 **switch** (flag) {  
 **case** 7:

afterList.add(str);  
 **break**;  
 **case** 1:  
 stack.push(str);  
 **break**;  
 **case** 2:  
 String pop=stack.pop();  
 **while**(!pop.equals(**"("**)){  
 afterList.add(pop);  
 pop=stack.pop();  
 }  
 **break**;  
 **default**:  
 **if**(stack.isEmpty()){  
 stack.push(str);  
 **break**;  
 }  
 **else**{  
 **if**(stack.peek().equals(**"("**)){  
 stack.push(str);  
 **break**;  
 }**else**{  
 **int** ji1=**this**.youxianji(str);  
 **int** ji2=**this**.youxianji(stack.peek());  
 **if**(ji1>ji2){  
 stack.push(str);  
 }**else**{  
 **while**(!stack.isEmpty()){  
 String f=stack.peek();  
 **if**(f.equals(**"("**)){  
 stack.push(str);  
 **break**;  
 }**else**{  
 **if**(**this**.youxianji(str)<=**this**.youxianji(f)){  
 afterList.add(f);  
 stack.pop();  
 }**else**{  
 stack.push(str);  
 **break**;  
 }  
 }

}  
 **if**(stack.isEmpty()){  
 stack.push(str);  
 }  
 }  
 **break**;  
 }  
 }  
 }  
 }  
 **while**(!stack.isEmpty()){  
 afterList.add(stack.pop());  
 }  
 StringBuffer sb=**new** StringBuffer();  
 **for**(String s:afterList){  
 sb.append(s+**" "**);  
 }  
 *//System.out.println(sb.toString());* **return** afterList;  
 }  
 */\*\*  
 判断运算符的优先级  
 \*/* **public int** youxianji(String str){  
 **int** result=0;  
 **if**(str.equals(**"+"**)||str.equals(**"-"**)){  
 result=1;  
 }**else**{  
 result =2;  
 }  
 **return** result;  
 }  
 */\*\*  
 \*判断字符串属于操作数、操作符还是括号  
 \*/* **public int** matchWitch(String s){  
 **if**(s.equals(**"("**)){  
 **return** 1;  
 }**else if**(s.equals(**")"**)){  
 **return** 2;  
 }**else if**(s.equals(**"+"**)){  
 **return** 3;  
 }**else if**(s.equals(**"-"**)){

**return** 4;  
 }**else if**(s.equals(**"\*"**)){  
 **return** 5;  
 }**else if**(s.equals(**"/"**)){  
 **return** 6;  
 }**else**{  
 **return** 7;  
 }  
 }  
 */\*\*  
 \*计算a@b的简单方法  
 \*/* **public** Double singleEval(Double pop2,Double pop1,String str){  
 Double value=0.0;  
 **if**(str.equals(**"+"**)){  
 value=pop2+pop1;  
 }**else if**(str.equals(**"-"**)){  
 value=pop2-pop1;  
 }**else if**(str.equals(**"\*"**)){  
 value=pop2\*pop1;  
 }**else** {  
 value=pop2/pop1;  
 }  
 **return** value;  
 }  
 **private double result**;  
 **public double** getResult() {  
 **return result**;  
 }  
 **public void** setResult(**double** result) {  
 **this**.**result** = result;  
 }  
 **private int state**;  
  
 **public int** getState() {  
 **return state**;  
 }  
 **public void** setState(**int** state) {  
 **this**.**state** = state;  
 }  
 **public void** countHouzhui(List<String> list){  
 **str** = **new** StringBuilder(**""**);

**state**=0;  
 **result**=0;  
 Stack<Double> stack=**new** Stack<Double>();  
 **for**(String str:list){  
 **int** flag=**this**.matchWitch(str);  
 **switch** (flag) {  
 **case** 3:  
 **case** 4:  
 **case** 5:  
 **case** 6:  
 Double pop1=stack.pop();  
 Double pop2=stack.pop();  
 Double value=**this**.singleEval(pop2, pop1, str);  
 stack.push(value);  
 **break**;  
 **default**:  
 Double push=Double.*parseDouble*(str);  
 stack.push(push);  
 **break**;  
 }  
 }  
 **if**(stack.isEmpty()){  
 **state**=1;  
 }**else**{  
 **result**=stack.peek();  
 **str**.append(stack.pop());  
 }  
 }  
 **public void** eval()**throws** Exception{  
 List<String> list=**new** ArrayList<String>();  
 *//匹配运算符、括号、整数、小数，注意-和\*要加\\* Pattern p = Pattern.*compile*(**"[+\\-/\\\*()]|\\d+\\.?\\d\*"**);  
 Matcher m = p.matcher(**s1**);  
 **while**(m.find()){  
 list.add(m.group());  
 }  
 List<String> afterList=**this**.midToAfter(list);  
 **this**.countHouzhui(afterList);  
 }  
 }  
}

## 3.3活动设计

运行过程基础分析如图下：

图3.3运行分析图

计算器

基本的加减乘除运算

加法

除法

乘法

减法

# 第4章、系统测试与部署

## 4.1项目开发文件夹：

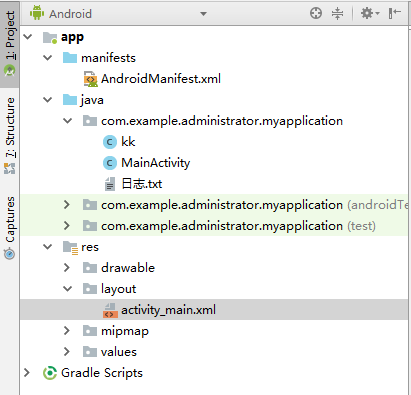


图4.1项目开发文件夹

## 4.2系统运行界面：

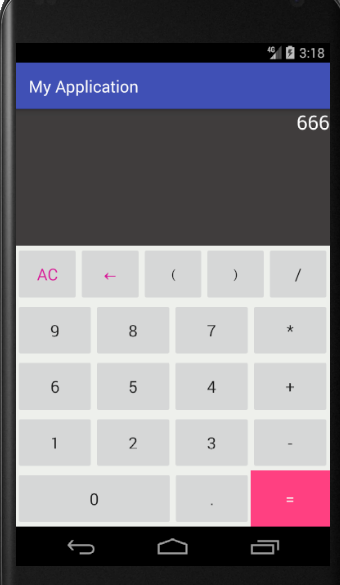


图4.2系统运行界面

# 第5章、实训总结

通过本次的实训，我感受颇多，一开始面对这个实训题目时，除了设计主界面，其他的一些功能什么的感觉无从下手。到实践应用中，常因为一点小错误而前功尽弃。

当然过程也需要美化，怎样让手中的素材尽可能地动起来，从而吸引别人的眼球。那就必须充分运用教程中所学到的制作技巧，举一反三，大胆尝试，在学习阶段要尽量把老师所教的内容充分运用到作品中去，这样才能学深学透。遇有不懂的地方要随时请教同学、老师，要做到不耻下问，好在我每天都有和同学交流，那样我就能够在欣赏别人作品的同时也能让自己有一个灵感，使自己的程序更加美观。同时，在设计过程中一定要耐心、细致，对于其中的细节、布局等方面要反复推敲、练习，考虑周祥，力求做到完美。

在整个过程中，我感觉就像人生的一个缩影，充满了酸甜苦辣。当一个问题我想了好久也没办法解决时，我感到沮丧与无助；当经过努力解决了一个程序上的难题时，我感到无比的喜悦；当经过每天为了写代码，找素材一坐就是一上午甚至一天的时候，我感到无比的痛苦。但艰难痛苦已成往事，灿烂的明天正向我们走来，今天我们总算迎来了胜利的坚果，以前的辛酸和艰苦化作了我们坚强的力量，将在我们将来的人生历程中，为我们的美好明天打下坚实的基础。经过这次的实训，我又一次的感受到了只有付出才会有收获，成功永远只属于不懈努力的人。

人非生而知之，要学得知识，一靠学习，二靠实践。没有实践，学习就是无源之水，无本之木。