**学 堂 在 线**

**“移动互联网开发”课程报告**

**学生姓名： 张承楷**

**学生学号： 3180611023**

**注册邮箱： 1468677239@qq.com**

**2020年11月21日**

目录

[一、开发任务 1](#_Toc519000864)

[二、开发环境 1](#_Toc519000865)

[三、系统设计 1](#_Toc519000866)

[3.1页面与功能设计 1](#_Toc519000867)

[3.2数据库设计 2](#_Toc519000868)

[四、运行结果 2](#_Toc519000869)

[五、主要源码及介绍 3](#_Toc519000870)

[5.1、MainActivity.java代码 3](#_Toc519000871)

[5.2、CountHelper.java类（数据库连接与操作） 8](#_Toc519000872)

[5.3、ResultViewActivity.java（历史记录显示） 9](#_Toc519000873)

[参考文献 11](#_Toc519000874)

# 一、开发任务

开发的内容为简便计算器设计，包括以下内容：

（1）包含两个Activity，第一个Activity为计算器主界面，主要功能是进行数据的计算；第二个Activity是历史记录页面，主要功能是显示历史计算记录。

（2）该计算器可以进行2个数字的加、减、乘、除运算。

（3）运算结果可以用sqlite数据库进行记录，然后在第二个Activity上以Listview的形式显示出来。

# 二、开发环境

开发采用Android Studio。它包括编译器、资源管理器、调试器和模拟器。 JDK环境为JDK8版本，Android Studio版本为4.1。

# 三、系统设计

## 3.1页面与功能设计

整个程序基于android技术开发，出总体模块外主要分为输入模块、显示模块以及计算模块（包括一些其她功能）这三大部分。输入模块部分负责读取用户输入的数据，显示模块部分负责显示用户之前的数据以及显示最终的计算结果，计算模块部分负责进行数据的运算规则及相应的结果。

输入模块主要描述了计算器键盘以及键盘的监听即主要负责读取用户的键盘输入以及响应触屏的按键，需要监听手机按键动作以及用指针事件处理方法处理触屏的单击动作；同时提供较为直观的键盘图形用户界面。

显示模块描述了计算器的显示区，即该区域用于显示用户输入的数据以及最终的计算结果；同时还负责显示一些其它信息。

计算模块则主要描述计算器的整体功能，实现计算器的页面，负责用户输入数据，包括：加、减、乘、除、等各种计算功能，记忆数据的相关功能，退格以及清零功能。

如图1所示是计算器的功能结构图。计算器监听键盘的输入，实时显示输入的数字及运算规则；当使用者选择清零和退格操作时，计算器所有输入框全部清空并等待键盘再次输入；当使用者点击计算按钮使，根据输入的数据和计算规则计算最终的计算数据，并在相应位置显示最终结果；当使用者点击跳转按钮时，跳转到历史记录页面，查看所有的计算结果。



图1 计算器功能流程图

## 3.2数据库设计

历史记录显示的信息包括计算次数的id、计算规则、计算结果等三项数据信息，故本项目设计了如下SQLite的数据库：

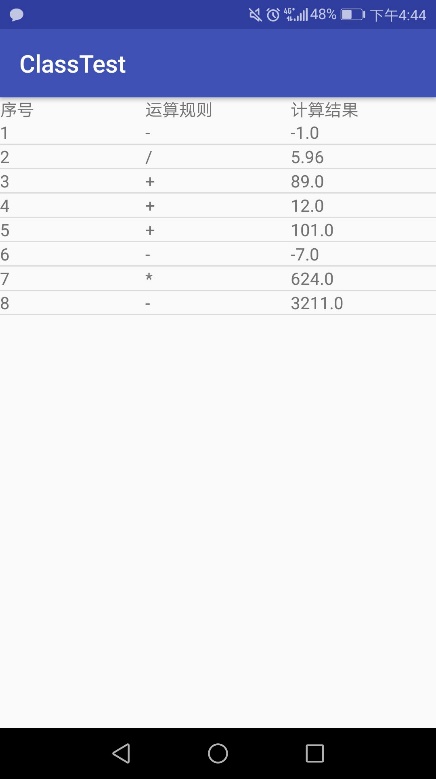
表1 SQL数据库各字段设置

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 是否主键 | 说明 |
| \_id | integer | 是 | id，自增加 |
| \_operator | varchar | 否 | 运算符 |
| \_result | varchar | 否 | 运算结果 |

# 四、运行结果

根据上述的设计，本次设计开发了一套功能完善的计算器，下面对页面及功能进行详细的介绍和讲解：

如图2（1）所示是整个计算器的主页面，包括：显示模块和计算模块。显示模块显示使用者输入的数据、计算法则和计算所得的结果。图2（2）是计算结果测试页面，测试实例的输入数据为4063和852，计算两者差值，得到结果3211，计算结果正确。图2（3）是历史记录页，记录之前所有的计算结果，包括计算序号、计算法则和计算所得结果。

（1）主页面 （2）测试计算 （3）历史记录

图2 程序运行截图

# 五、主要源码及介绍

## 5.1、MainActivity.java代码

（1）定义相应控件名

1. TextView number1\_text,numberY\_text;,numberResult\_text;
2. Button button0,button1, button2,button3,button4,button5,button6,button7,button8,button9,
3. button\_plus,button,button,buttonCU,buttonPoint,buttonQY,buttonDelect,buttonClean,buttonCount,buttonPast;
4. int YSF;                             //用于计算法则
5. int number1 = 0;
6. int number2 = 0;
7. Double numberResult;
8. Boolean stop = false;                   //是否结束输入，判断数据的分割点
9. private CountHelper mHelper;

（2）设定相应的数字识别监听机制，根据不同的按钮识别不同的数字输入和计算法则输入

1. button0.setOnClickListener(**new** View.OnClickListener() {
2. **public** **void** onClick(View view) {
3. **if** ( !stop) {
4. number1 = 10\*number1 + 0;
5. number1\_text.setText(String.valueOf(number1));
6. } **else** {number2 = 10\*number2 +0;
7. number2\_text.setText(String.valueOf(number2));
8. }}});

（3）计算法则

1. 加法：
2. button\_plus = (Button) findViewById(R.id.button\_add);//加
3. //加法按钮
4. button\_plus.setOnClickListener(**new** View.OnClickListener() {
5. **public** **void** onClick(View view) {
6. numberY\_text.setText("+");
7. YSF = 1;stop = **true**; }});
8. 减法：
9. buttonJ = (Button) findViewById(R.id.button\_j);//建
10. //减法按钮点击
11. buttonJ.setOnClickListener(**new** View.OnClickListener() {
12. @Override
13. **public** **void** onClick(View view) {
14. numberY\_text.setText("-");
15. YSF = 2;
16. stop = **true**;
17. }
18. });
19. 乘法：
20. buttonC = (Button) findViewById(R.id.button\_c);//乘法
21. buttonC.setOnClickListener(**new** View.OnClickListener() {
22. @Override
23. **public** **void** onClick(View view) {
24. numberY\_text.setText("\*");
25. YSF = 3;
26. stop = **true**;
27. }
28. });
29. 除法：
30. buttonCU = (Button) findViewById(R.id.button\_chu);//除
31. buttonCU.setOnClickListener(**new** View.OnClickListener() {
32. @Override
33. **public** **void** onClick(View view) {
34. numberY\_text.setText("/");
35. YSF = 4;
36. stop = **true**;
37. }
38. });

（4）计算过程

1. buttonCount.setOnClickListener(**new** View.OnClickListener() {
2. @Override
3. **public** **void** onClick(View view) {
4. **if**(YSF == 1) {
5. numberResult = (**double**)number1 + number2;
6. numberResult\_text.setText(String.valueOf(numberResult));
7. stop = **false**;number1 = 0;number2 = 0;YSF = 0;
8. SQLiteDatabase db = mHelper.getWritableDatabase();
9. ContentValues values = **new** ContentValues();
10. values.put("yun",String.valueOf(1));
11. values.put("cr",String.valueOf(numberResult));
12. **long** newRowId = db.insert("a",**null**,values);
13. } **else** **if**(YSF==2) {
14. numberResult = (**double**)number1 - number2;
15. numberResult\_text.setText(String.valueOf(numberResult));
16. stop = **false**;number1 = 0;number2 = 0;YSF = 0;
17. SQLiteDatabase db = mHelper.getWritableDatabase();
18. ContentValues values = **new** ContentValues();
19. values.put("yun",String.valueOf(2));
20. values.put("cr",String.valueOf(numberResult));
21. **long** newRowId = db.insert("a",**null**,values);
22. } **else** **if**(YSF == 3) {
23. numberResult = (**double**)number1 \* number2;
24. numberResult\_text.setText(String.valueOf(numberResult));
25. stop = **false**;number1 = 0;number2 = 0;YSF = 0;
26. SQLiteDatabase db = mHelper.getWritableDatabase();
27. ContentValues values = **new** ContentValues();
28. values.put("yun",String.valueOf(3));
29. values.put("cr",String.valueOf(numberResult));
30. **long** newRowId = db.insert("a",**null**,values);
31. }**else** **if**(YSF == 4) {
32. numberResult = (**double**)number1 / number2;
33. numberResult\_text.setText(String.valueOf(numberResult));
34. stop = **false**;
35. number1 = 0;
36. number2 = 0;
37. YSF = 0;
38. SQLiteDatabase db = mHelper.getWritableDatabase();
39. ContentValues values = **new** ContentValues();
40. values.put("yun",String.valueOf(4));
41. values.put("cr",String.valueOf(numberResult));
42. **long** newRowId = db.insert("a",**null**,values);
43. }**else** {
44. Toast.makeText(MainActivity.**this**,"输入有误！
45. ",Toast.LENGTH\_SHORT).show();
46. stop = **false**;
47. number1 = 0;
48. number2 = 0;
49. }
50. }
51. });

（5）退格功能

1. buttonDelect = (Button) findViewById(R.id.button\_delect);
2. buttonDelect.setOnClickListener(**new** View.OnClickListener() {
3. @Override
4. **public** **void** onClick(View view) {
5. **if**(number2!=0){
6. number2\_text.setText("");
7. number2 = 0;
8. }
9. **else** **if**(number2==0){
10. **if**(YSF==0){
11. number1\_text.setText("");
12. number1 = 0;
13. }
14. **else** {
15. numberY\_text.setText("");
16. YSF = 0;
17. stop=**false**;
18. }
19. }
20. }
21. });

（6）清除功能

1. buttonDelect = (Button) findViewById(R.id.button\_delect);
2. buttonDelect.setOnClickListener(**new** View.OnClickListener() {
3. **public** **void** onClick(View view) {
4. **if**(number2!=0){
5. number2\_text.setText("");
6. number2 = 0;}

## 5.2、CountHelper.java类（数据库连接与操作）

（1）创建数据库

1. **public** **void** onCreate(SQLiteDatabase sqLiteDatabase) {
2. String create\_tablea = "create table if not exists " + "a" + "("
3. +"\_id" + " integer primary key autoincrement,"
4. +"yun" + " text,"+"cr" + " text)";
5. sqLiteDatabase.execSQL(create\_tablea);}

（2）更新数据库

1. SQLiteDatabase db = mHelper.getWritableDatabase();
2. ContentValues values = **new** ContentValues();
3. values.put("yun",String.valueOf(3));
4. values.put("cr",String.valueOf(numberResult));
5. **long** newRowId = db.insert("a",**null**,values);

## 5.3、ResultViewActivity.java（历史记录显示）

（1）数据库连接与列表建立

1. ListView listView;
2. CountHelper mHelper;
3. SQLiteDatabase db;
4. ArrayList<HashMap<String,Object>> listItem;
5. SimpleAdapter simpleAdapter;
6. **private** Cursor cursor1;
7. listView = (ListView) findViewById(R.id.listview);
8. mHelper = **new** CountHelper(**this**);
9. db = mHelper.getReadableDatabase();
10. String[] projection = {"\_id","yun","cr"};
11. listItem = **new** ArrayList<HashMap<String, Object>>();

（2）数据库查询与显示

1. **int** idColumnIndex = cursor1.getColumnIndex("\_id");
2. **int** titleColumnIndex = cursor1.getColumnIndex("yun");
3. **int** timeColumnIndex = cursor1.getColumnIndex("cr");
4. **while** (cursor1.moveToNext()) {
5. HashMap<String,Object> map = **new** HashMap<String, Object>();
6. **int** currentID = cursor1.getInt(idColumnIndex);
7. String ID = String.valueOf(currentID);
8. String t = cursor1.getString(titleColumnIndex);
9. String re = cursor1.getString(timeColumnIndex);
10. **if**(t.equals(String.valueOf(1))) {
11. tt = "+";
12. }**else** **if**(t.equals(String.valueOf(2))){
13. tt = "-";
14. }**else** **if**(t.equals(String.valueOf(3))){
15. tt = "\*";
16. }**else** **if**(t.equals(String.valueOf(4))){
17. tt = "/";
18. }
19. map.put("id",ID);
20. map.put("title" , tt);
21. map.put("time" , re);
22. listItem.add(map);
23. simpleAdapter = **new** SimpleAdapter(**this**,listItem,R.layout.item\_layout,
24. **new** String[]{"id","title","time"},
25. **new** **int**[]{R.id.number\_result,R.id.yun\_result,R.id.jie\_result});
26. listView.setAdapter(simpleAdapter);
27. } **finally** {
28. cursor1.close();
29. }
30. }

# 参考文献

[1]姚昱旻,刘卫国.Android的架构与应用开发研究[J].计算机系统应用,2008(11):110-112+24.

[2]公磊,周聪.基于Android的移动终端应用程序开发与研究[J].计算机与现代化,2008(08):85-89.

[3]曾健平,邵艳洁.Android系统架构及应用程序开发研究[J].微计算机信息,2011,27(09):1-3.

[4]赵亮,张维.基于Android技术的界面设计与研究[J].电脑知识与技术,2009,5(29):8183-8185.

[5]张玉清,王凯,杨欢,方喆君,王志强,曹琛.Android安全综述[J].计算机研究与发展,2014,51(07):1385-1396.