

中山大学数据科学与计算机学院本科生实验报告

(2017 年秋季学期)

课程名称：手机应用平台开发

任课教师：刘宁

年级	15 级	专业 (方向)	软件工程数媒方向
学号	15331016	姓名	陈桂燕
电话	13719346314	Email	735594896@qq.com
开始日期	2017/12/4	完成日期	2017/12/5

一、 实验题目：数据存储 (二)

【目的】

1. 学习 SQL 数据库的使用；
2. 学习 ContentProvider 的使用；
3. 复习 Android 界面编程。

二、 实现内容

实现一个生日备忘录，要求实现：

1. 使用 SQLite 数据库保存生日的相关信息，并使得每一次运行程序都可以显示出已经存储在数据库里的内容；
2. 使用 ContentProvider 来获取手机通讯录中的电话号码。

三、实验过程

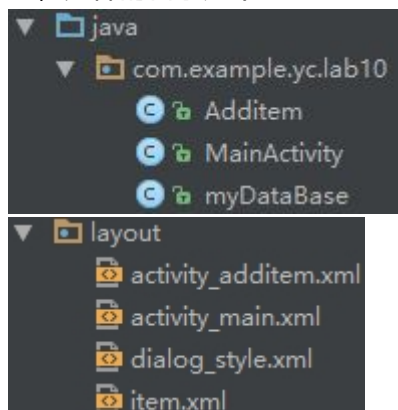
1. 前提



主界面 activity_main.xml → 点击“增加条目”进入编辑界面 activity_additem.xml

UI 设计里，控件的实现基本遵循实验要求，在此基础上自己制作了背景图并改变了颜色搭配。ListView 的 item.xml 样式设计里，姓名栏采取“居左对齐”，生日栏采取“居中对齐”，“礼物栏”采取居右对齐。

工程文件的目录如下：



2. SQLite 数据库

首先我建立了一个 java 类 : myDataBase.java , 然后在里面使用 SQLiteOpenHelper 的子类新建了一个数据库的类 , 代码如下。利用 onCreate , 根据存储需求(每一条目的 name、birthday、gift) 创建了数据库 , 再重写 onUpgrade 函数 , 由于本次实验不必对数据进行更新升级 , 所以 onUpgrade 函数里面什么也没写。

```
public class myDataBase extends SQLiteOpenHelper {
    private static final String DB_NAME = "MyDB.db";
    private static final String TABLE_NAME = "Info";
    private static final int DB_VERSION = 1;

    public myDataBase(Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    public myDataBase(Context context) { this(context, DB_NAME, null, DB_VERSION); }

    // 创建数据库, 直接执行SQL语句即可
    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
        String CREATE_TABLE = "Create Table "
            + TABLE_NAME
            + "(_id integer primary key , "
            + "name text, "
            + "birth text, "
            + "gift text);";
        sqLiteDatabase.execSQL(CREATE_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    }
}
```

然后在 MainActivity.java 里 , 主要执行的函数如下。每次对数据进行增删改的操作后都应再次执行 init () , 及时更新数据。

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

    findViewById();
    init();//对列表里应该呈现的数据信息进行更新
    myClick();
}
```

Listview 的设置和之前教的一样 , 首先为 Listview 设计专门的样式 xml 文件 , 然后利用 Map 组成的 List 来储存数据 , 再利用相关的参数来创建适配器 SimpleAdapter。不同的是 , 之前 List 的初始数据获取为手动添加 , 而这次是通过遍历数据库的方式来获取数据的。

```
private ListView listView;
private SimpleAdapter simpleAdapter;
private List<Map<String, String>> data;
private myDataBase db;
private SQLiteDatabase sqLiteDatabase;
private Cursor cursor;
```

所用到的参数如上 , 通过下面的 init () 函数 , 对列表数据进行刷新并呈现 ↓

```

void init() {
    db = new myDataBase(getBaseContext());
    SQLiteDatabase = db.getWritableDatabase();
    cursor = SQLiteDatabase.rawQuery("select * from Info", null); 查询全部
    data = new ArrayList<>();
    if (cursor != null) {
        while (cursor.moveToNext()) { 循环获取
            String name_ = cursor.getString(1);
            String birth_ = cursor.getString(2);
            String gift_ = cursor.getString(3);
            Map<String, String> map = new HashMap<>();
            map.put("name", name_);
            map.put("birth", birth_);
            map.put("gift", gift_);
            data.add(map);
        }
        simpleAdapter = new SimpleAdapter(MainActivity.this, data, R.layout.item,
            new String[]{"name", "birth", "gift"}, new int[]{R.id.Name, R.id.Birth, R.id.Gift});
        listView.setAdapter(simpleAdapter);
    }
}

```

>> SQLite 数据的“查”和“增” ↓

在 Additem.java 对应的编辑界面里，填写信息后点击“增加”按钮，名字不为空后，通过查询操作确定 name 是否存在，若不存在再对数据库进行数据增加，代码如下。

```

void myClick() {
    add_confirm.setOnClickListener((view) -> {
        myDataBase db = new myDataBase(getBaseContext());
        SQLiteDatabase = db.getWritableDatabase();
        Cursor cursor = SQLiteDatabase.rawQuery("select * from Info where name like ?", new String[]{name_Text. 查询该名字是否存在

        if (name_Text.getText().toString().equals("")) {
            Toast.makeText(Additem.this, "名字为空，请完善", Toast.LENGTH_SHORT).show();
        } else if (cursor.moveToFirst()) {
            Toast.makeText(Additem.this, "名字重复啦，请检查", Toast.LENGTH_SHORT).show();
        } else {
            ContentValues contentValues = new ContentValues();
            contentValues.put("name", name_Text.getText().toString());
            contentValues.put("birth", birth_Text.getText().toString());
            contentValues.put("gift", gift_Text.getText().toString());
            SQLiteDatabase.insert("Info", null, contentValues); 数据增加
            SQLiteDatabase.close();
            setResult(1, new Intent());
            finish();
        }
    });
}

```

>> SQLite 数据的“删” ↓

在 MainActivity.java 对应的主界面里，长按列表某一项会弹出“是否删除”的 AlertDialog。选择“是”后进行数据的删除。（直接通过 execSQL 即可，同时数据也要从列表 List 移除）

```

.setPositiveButton("是", (dialogInterface, i) -> {
    db = new myDataBase(getBaseContext());
    SQLiteDatabase = db.getWritableDatabase();
    SQLiteDatabase.execSQL("DELETE FROM Info WHERE name = ?", new String[]{data.get(arg).get("name")});
    data.remove(arg);
    simpleAdapter.notifyDataSetChanged();
});

```

>> SQLite 数据的“改” ↓

在 MainActivity.java 主界面里，点击列表某一项会弹出对应的信息表，除名字不可改外，其他信息进行修改后点击“保存修改”，通过指定 name 对应的数据项而对其他变量进行改动。（直接通过 execSQL 即可，并及时更新数据）

```

db = new myDataBase(getBaseContext());
SQLiteDatabase = db.getWritableDatabase();
SQLiteDatabase.execSQL("update Info set birth = ? where name = ?", new Object[]{new_birth, old_name});
SQLiteDatabase.execSQL("update Info set gift = ? where name = ?", new Object[]{new_gift, old_name});
SQLiteDatabase.close();
init();

```

3. ContentProvider 的使用

首先在 manifest 里进行申明。最开始我不小心把 uses 写出 user 了，查了好久才发现.....

~~user~~-permission android:name = "android.permission.READ_CONTACTS"/>
在手机上运行时，要先在该 app 的权限管理里允许“读取联系人”，不然每次点击列表就会出现闪退的情况。

读取联系人
读取联系人



然后每次点击列表项，信息里电话那栏便要进行联系人电话的读取。这里有点像前面的数据库读取，先是通过 query 查询全部再逐一循环，然后判断联系人姓名是否相同，若相同便再获取对应的号码。代码如下：

```
//读取联系人电话
String find_tel = "";
cursor = getContentResolver().query(ContactsContract.Contacts.CONTENT_URI, null, null, null, null);
while (cursor.moveToNext()) {
    String str = cursor.getString(cursor.getColumnIndex("_id"));
    if (cursor.getString(cursor.getColumnIndex("display_name")).equals(Name.getText().toString()) {
        if (Integer.parseInt(cursor.getString(cursor.getColumnIndex("has_phone_number"))) > 0) {
            Cursor tcursor = getContentResolver().query(
                ContactsContract.CommonDataKinds.Phone.CONTENT_URI, null, "contact_id = " + str, null, null);
            while (tcursor.moveToNext()) {
                find_tel = find_tel + tcursor.getString(tcursor.getColumnIndex("data1"));
            }
            tcursor.close();
        }
    }
}
cursor.close();
if(find_tel.equals(""))
    find_tel = "无";
telephone.setText(find_tel);
```

4. 其他部分

(1) 点击列表项会弹出可以修改的信息栏，样式如下：



可以看到，这不是传统的 AlertDialog 样式。实现如下：

首先设计一个界面 xml，然后再使用 LayoutInflater 实现，代码如下。

```
//创建一个自定义的dialog，里面包含信息
public void showDialog(Context context, Map<String, String> map) {
    View view = LayoutInflater.from(context).inflate(R.layout.dialog_style, null);
    final Dialog dialog = new Dialog(context);
    dialog setContentView(view);
}
```

为了使其显示效果更好，可以自己设置 dialog 的大小。

```
dialog.show();
//设置dialog的大小
Window window = dialog.getWindow();
WindowManager.LayoutParams lp = window.getAttributes();
lp.gravity = Gravity.CENTER;
lp.width = 920;
lp.height = 820;
dialog.getWindow().setAttributes(lp);
```

(2) 在 MainActivity 主界面点击“增加条目”按钮后，通过 startActivityForResult 进入 Additem 编辑界面，填写信息确认添加后 setResult 并结束该 intent 返回主界面，记得要在 MainActivity 里 onActivityResult 函数里执行 init()，及时刷新列表。

(其实每次对数据库 SQLite 做相关操作后都要记得更新数据，我都是通过执行 init()实现)

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent intentData) {
    if (requestCode == 1) {
        if (resultCode == 1) {
            init();
        }
    }
}
```

四、实验思考及感想

由于之前我们小组的期中项目“三国辞典”便是通过 SQLite 来实现“增删改查”的功能的，当时为了实现功能所以就查阅了 SQLite 相关使用方法，因此这次实验在数据库这方面开展得比较顺利。然后 contentprovider 那里，刚开始没仔细看实验文档，所以不知道要先允许“读取联系人”权限，每次点击列表项都直接整个 app 闪退，还以为是自己的自定义对话框设置哪里出了 bug。实验文档提供的信息都是比较有价值的，以后会更加仔细阅读。

另外，关于 ListView 适配器的一系列代码，虽然已经学过了，但是让自己重新敲打突然就无从下手了，还是得翻出之前实验打过的代码进行“参考模仿”。记忆力可以说是很差劲了。