

实验八

数据存储（二）

【实验目的】

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

【实验内容】

实现一个生日备忘录，

技术要求：

- 1、使用 SQLite 数据库保存生日的相关信息，使得每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 来获取对应寿星的电话号码；

功能要求：

- 1、主界面包含增加生日条目按钮和生日信息列表；（见图 1，图 2）
- 2、点击<增加条目>按钮跳转到次界面；
- 3、次界面输入生日相关信息后点击<增加>按钮会返回主界面（同时更新主界面的生日信息列表），且姓名字段不能为空，姓名字段不能重复（见图 3，图 4）
- 4、主界面中的列表点击事项处理：
 - a) 单击（查看并可修改该生日条目）：（见图 5）
 - i. 弹出对话框，显示该条目的相关信息，并提供修改。
 - ii. 同时，显示该生日条目寿星的电话号码；
 - iii. 点击<保存修改>按钮，更新主界面的生日信息列表
 - b) 长按（可删除该生日条目）：（见图 6）
 - i. 弹出对话框，显示是否删除；
 - ii. 点击<是>按钮，删除该生日条目，并更新主界面的生日信息列表



图 1 首次启动



图 2 增加一些条目后



图 3 名字不能重复



图 4 名字不能为空



图 5 点击处理



图 6 长按处理

【参考内容】

SQLite 数据库使用

使用 SQLiteOpenHelper 的子类能更方便实现要求:

1、创建类

```
public class myDB extends SQLiteOpenHelper {
    private static final String DB_NAME = "Re
    private static final String TABLE_NAME =
    private static final int DB_VERSION = 1;
```

2、创建数据库：可直接执行创建数据库的 SQL 语句

```

@Override
public void onCreate(SQLiteDatabase sqLiteDatabase) {
    String CREATE_TABLE="CREATE TABLE if not exists "
        +TABLE_NAME
        +" (_id INTEGER PRIMARY KEY,name TEXT,birth TEXT,gift TEXT)";
    sqLiteDatabase.execSQL(CREATE_TABLE);
}

```

3、onUpgrade 在本次实验不需要用到，但 Android 要求重写才能实例化。

```

@Override
public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int il) {

}

```

4、实现增加、更新和删除这 3 种操作有两种方法：

不管是哪种方法，记得先 getWritableDatabase ()

(a) 用 execSQL 方法直接执行相应的 SQL 语句，比如增加（如下）。

```

SQLiteDatabase db = getWritableDatabase();
String insert_sql="INSERT INTO <表名>(<列 1>,<列 2>,...)
values(<值 1>,<值 2>,...)";
db.execSQL(insert_sql);

```

(b) 使用相应的 insert、update 和 delete 方法

I. insert 方法需要使用 ContentValues 来存放要添加的数据，见下图

```

public void insert2DB(String name,String birth,String gift){
    SQLiteDatabase db = getWritableDatabase();
    ContentValues cv=new ContentValues();
    cv.put("name",name);
    cv.put("birth",birth);
    cv.put("gift",gift);
    db.insert(TABLE_NAME,null,cv);
    db.close();
}

```

II. update 方法需要使用 ContentValues 和 Where 语句。（下图只是说明性代码）

```

 SQLiteDatabase db = getWritableDatabase();
 ContentValues cv=new ContentValues();
 cv.put("<列1>", "<值1>");
 cv.put("<列2>", "<值2>");
 //...
 String whereClause="<主键列名>=? ";
 String[] whereArgs={"<?的值>"};

 db.update(TABLE_NAME, cv, whereClause, whereArgs);

 //如果删除的主键列值只有一个, 也可以这样写
 String whereClause1="<主键列名>=<目标值>";
 db.update(TABLE_NAME, cv, whereClause1, null);
 db.close();

```

III、delete 方法需要使用 where 语句。(下图只是说明性代码)

```

 SQLiteDatabase db = getWritableDatabase();

 String whereClause="<主键列名>=? ";
 String[] whereArgs={"<?的值>"};

 db.delete(TABLE_NAME, whereClause, whereArgs);
 db.close();

```

5、实现查询操作可以使用 rawQuery 或 query 函数，它们的区别类似于上面，前者直接执行 SQL 语句，后者是通过参数组合产生 SQL 语句。(下图只是说明性代码)

进行查询前，记得先 getReadableDatabase()

```

 Cursor cursor = db.query(TABLE_NAME,
    new String[] {"<列1>", "<列2>", ...},
    null, null, null, null, null);

```

```

 Cursor cursor2 = db.rawQuery("select * from <TABLE_NAME>", null);

```

rawQuery 或者 query 函数返回的都是 Cursor，关于 Cursor 类的详细介绍请看下面的链接：

<http://www.cnblogs.com/TerryBlog/archive/2010/07/05/1771459.html>

Content Provider 使用

1、在 AndroidManifest.xml 文件里声明读取通讯录的权限

```
<uses-permission android:name="android.permission.READ_CONTACTS"/>
```

2、使用 `getContentResolver` 方法读取联系人列表

```
Cursor cursor =  
getContentResolver().query(ContactsContract.Contacts.CONTENT_URI, null  
, null, null, null);
```

3、判断某条联系人的信息中，是否有电话号码。

```
int isHas =  
Integer.parseInt(cursor.getString(cursor.getColumnIndex(ContactsContr  
act.Contacts.HAS_PHONE_NUMBER)));
```

4、取出该条联系人信息中的电话号码

```
Cursor c =  
getContentResolver().query(ContactsContract.CommonDataKinds.Phone.CON  
TENT_URI, null, ContactsContract.CommonDataKinds.Phone.CONTACT_ID + "  
= " + id, null, null);  
while (c.moveToNext()) {  
    number +=  
c.getString(c.getColumnIndex(ContactsContract.CommonDataKinds.Phone.N  
UMBER)) + " ";  
}  
c.close();
```

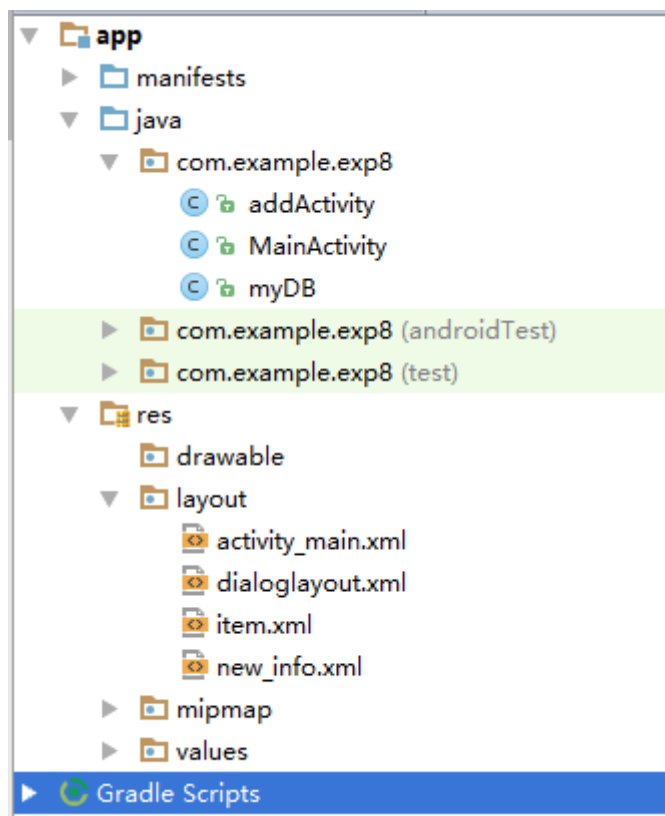
自定义的对话框实现

使用 `LayoutInflater` 类，如下：

```
LayoutInflater factory = LayoutInflater.from(MainActivity.this);  
View view = factory.inflate(R.layout.dialoglayout, null);  
AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);  
builder.setView(view);
```

如此之后，我们就可以将一个 `layout` 的内容（`dialoglayout`）全都显示在对话框（`builder`）里了。之后便可以对 `layout` 里面的元素进行操作了。

【参考工程目录结构】



【注意事项】

关于获取通讯录权限，在实际操作中，有些手机（比如：API 19 的）是在程序运行的时候进行询问是否给予权限；有些手机（比如：API 23 的），不会提示，需要你自己到设置界面下开启本应用程序的获取通讯录权限。

所以，如果你在点击表单的时候发生闪退，去看一眼获取通讯录的权限有没有开吧。

【检查内容】

是否完成所有实验内容的要求。

【提交说明】

- 1、Deadline：下一次实验课前一天晚上 12 点
- 2、提交前记得 clean 项目
- 3、提交位置：
 - 刘老师 <ftp://222.200.185.18:1890/> 对应文件夹
 - 郑老师 <ftp://my.ss.sysu.edu.cn/~zhgf> 对应文件夹

