**中山大学移动信息工程学院本科生实验报告**

**（2017年秋季学期）**

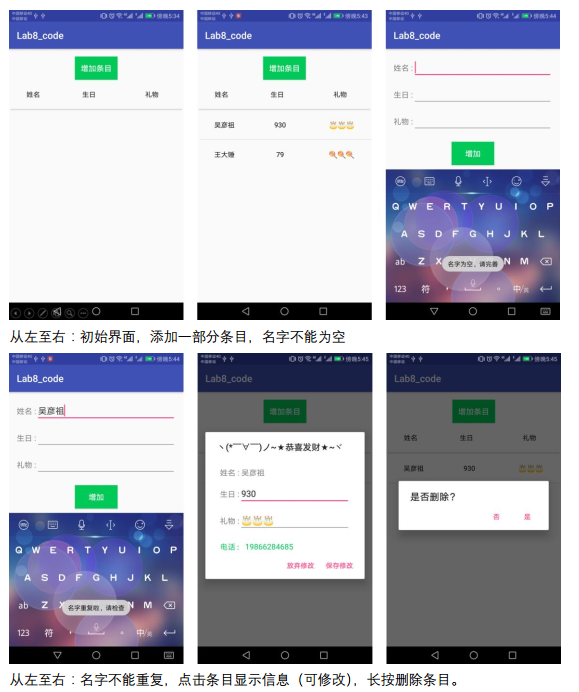
课程名称：移动应用开发 任课教师： 郑贵峰

|  |  |  |  |
| --- | --- | --- | --- |
| 年级 | 2015 | 专业（方向） | 移动信息工程（M2） |
| 学号 | 15352193 | 姓名 | 梁公勋 |
| 电话 | 15066299571 | Email | 1104695389@qq.com |
| 开始日期 | 2017.12.18 | 完成日期 | 2017.12.19 |

# 实验题目

数据存储（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。

功能要求：

A. 主界面包含增加生日条目按钮和生日信息列表；

B. 点击“增加条目” 按钮，跳转到下一个 Activity 界面， 界面中包含三个信息输入框（姓名、生日、

礼物） 和一个“增加” 按钮， 姓名字段不能为空且不能重复；

C. 在跳转到的界面中，输入生日的相关信息后， 点击“增加” 按钮返回到主界面， 此时，主界面中应

更新列表，增加相应的生日信息；

D. 主界面列表点击事件：

 点击条目：

弹出对话框，对话框中显示该条目的信息，并允许修改；

对话框下方显示该寿星电话号码（如果手机通讯录中有的话，如果没有就显示“无” ）

点击“保存修改” 按钮， 更新主界面生日信息列表。

 长按条目：

弹出对话框显示是否删除条目；

点击“是” 按钮，删除该条目，并更新主界面生日列表。

# 课堂实验结果

1. 实验截图

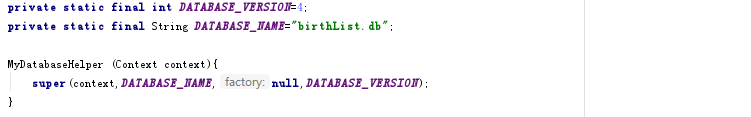
1. 实验步骤以及关键代码

**一、与数据库SQLite相关代码实现。**

这次的页面布局比较简单，也没有什么特别注意的点，所以这次就不在这里赘述布局文件了，直接开始解释java文件。

首先是与安卓的数据库有关的代码。安卓本身有一个数据库的封装类SQLiteOpenHelper，我们实现一个继承于这个类的子类MyDatabaseHelper用来管理APP的数据库。

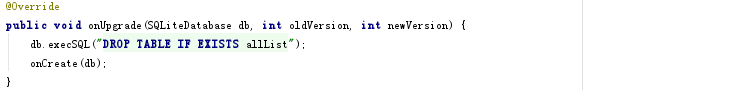
这个类的变量和构造函数如下所示。DATABASE\_VERSION是数据库版本，目前来看好像版本为1到4都可以用。DATABASE\_NAME为我们创建的数据库名称。在构造函数中，我们调用了super方法，传入了context、数据库名称和版本。这一句代码会直接为APP创建一个数据库，名称就是我们传入的名称。但是，我们在其他地方，会多次声明这个类，那按理说，每次声明都会创建一个数据库，但是APP和编译器都没有报错，所以应该是这个方法内部对这种重复声明的情况做了处理。



我们实现这个子类，还需要重载两个方法，其中一个是onCreate方法。我们在onCreate方法中创建了一个名为allList的表，用来存储我们的数据。这里需要调用数据库，使用其execSQL方法。SQLiteOpenHelper对于SQL语句实现的函数方法有很多。ExecSQL是最普遍的一条，即直接执行其形参中的语句。这里我们传入的语句就是创建一个allList表，这个表有name、birthday、gift列，类型均为TEXT，其中name为这个表的主键。



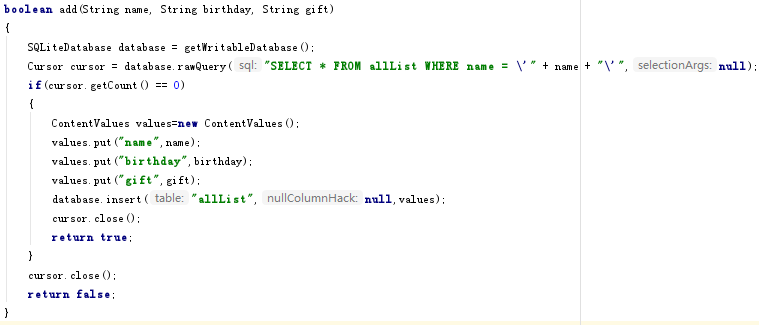
还需要重载的另一个方法是onUpgrade方法，这个方法在这次实验中没有用到，应该是负责数据库更新使用的，这里就是直接删掉旧表，然后调用onCreate方法重建新表。



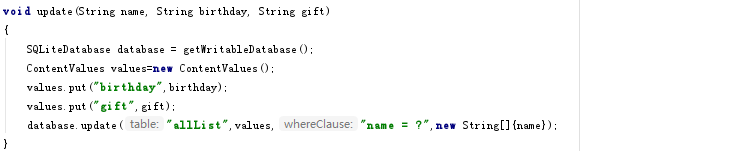
除了实现构造函数和必须重载的函数外，我们还可以在这个类中实现我们对于数据库的增删改查操作，以方便我们对于数据库的使用。

首先是往数据库中增加数据的操作，我们使用一个add函数实现这个操作。首先，我们需要使用getWritableDatabase函数获取我们要操作的数据库。这个方法指的是以可读写的方式打开数据库。这个是我们在对数据库的所有操作之前所必做的。获得数据库之后，我们调用数据库的rawQuery方法。这个方法是支持数据库查询操作的一个函数，有两个形参，第一个为查询语句，可以使用占位符，第二个形参便是占位符的字符串数组。这里没有使用占位符。这个方法会返回一个Cursor对象。这个对象是行的集合，其中的每一个对象都代表一行，我们必须知道某一列的名字以及类型，才能获取其某一列的数据。

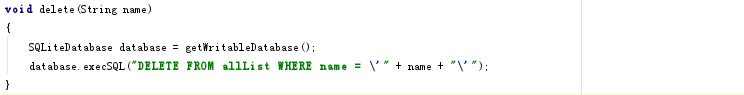
由于在表中，名字是主键，所以我们插入数据前必须查看主键是否已经存在。我们获取查询结果后，调用了cursor的getCount方法，获取cursor，也就是查询结果中 的记录数。如果记录数不为0，说明这个名称已经存在，返回false。如果为0，说明还没有这条记录，就进行插入。ContentValues是一种以键值对存储数据的对象，其中一个键值为列名，另一个为值，这里我们先将要插入的数据以键值对的方式按照列名插入其中。然后我们调用了数据库的inset方法插入数据，这个方法是专门实现数据库的插入操作的。其第一个形参为表名，第二个参数是为了防止插入列名为空的情况，第三个就是我们要插入的存储在ContentValues中的数据。最后需关闭cursor，然后返回真值。



接下来是实现了一个update函数实现数据库的数据更新操作。这里和插入数据有点相像。首先我妈妈也要获取数据库，然后使用contentValues对象将我们需要修改的数据以键值对的形式存入其中。最后调用数据库的update方法，进行数据更新。Update方法中有3个形参，第一个是要更新数据的表，第二个是条件，也就是SQL语句中where后的那些语句，这个也可以使用替代符，最后一个形参是替换替代符的字符串数组。



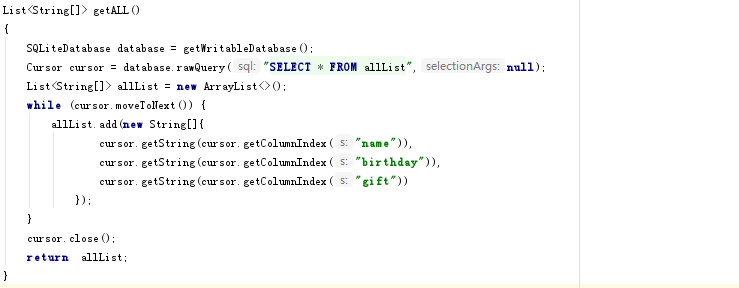
接下来是实现数据库的删除操作，这个是数据库中比较简单的语句。我们就获取数据库之后，直接调用执行普通SQL语句的execSQL方法，删除表中对应名字的记录即可。



最后是查询操作。数据库中，查询操作是最复杂的操作。为了实现SQL语句中查询语句的功能，SQLite重载了很多查询的方法。但这里我们不需要根据条件去查询，而只是取出所有记录将其显示出来即可，所以相对还算比较简单。

我们实现了getAll方法用来获取数据库中的所有记录。这个方法中，首先也是获取数据库，然后使用rawQuery方法查询数据。这个方法是查询操作中最简单的方法，只有两个形参，第一个是查询语句，第二个是替代符的字符集。

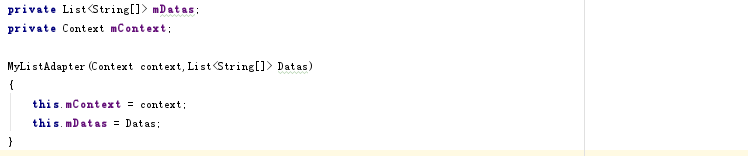
我们得到查询结果的cursor对象后，要逐行读取其数据。首先我们声明了一个字符串集的List来存储查询数据。Cursor的moveToNext函数是调到cursor下一行。然后根据cursor的列名和变量名，获取其这一行中每一列的值，并以字符集的形式添加到List中。所有的结果提取完毕后，关闭cursor，返回这个List。



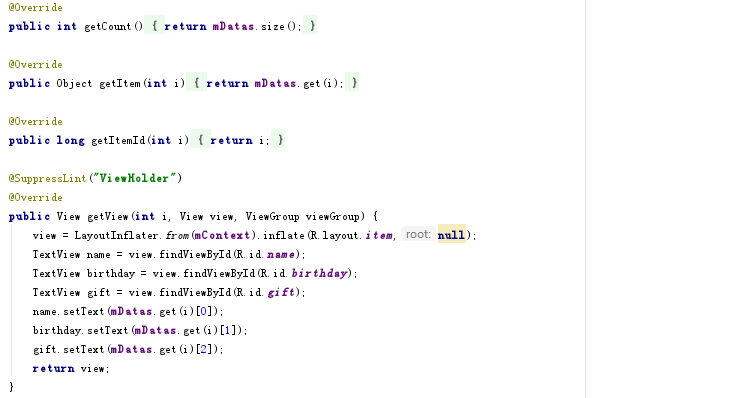
**二、自定义ListView的adapter。**

在主界面中，我们要显示每一个人的信息，还是需要使用ListView来展示。为了方便ListView的使用，我们还是自定义了一个adapter。这里采取了最简单，最低效的实现方法。

我们自定义的MyListAdapter类是继承于BaseAdapter类的。构造函数和变量如下所示。一个是要显示数据的List，也是字符串集类型的。然后还有一个Context变量。构造函数就是为这两个变量根据形参进行赋值。



为了实现这个自定义类，我们还必须实现必须重载的几个方法。这几个方法如下所示，其中最重要的是getView方法。在这个方法中，我们首使用inflate方法获取我们每一项显示的布局。然后用id获取这个布局中的控件。接着使用List中字符集的数据，为这些控件设置要显示的文本。最后就是返回这个View。



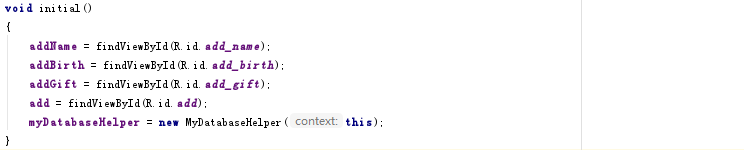
**三、增加条目的activity的相关代码**

我们点击主界面的增加条目后，会跳转到一个增加条目的activity（AddItem）。这里解释一下这个activity。

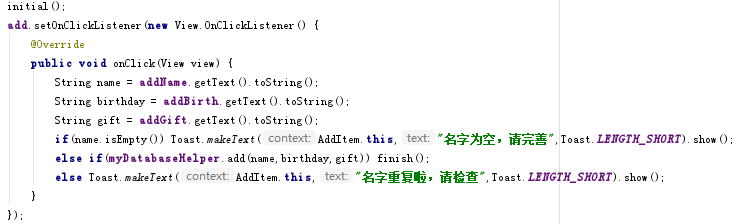
首先是变量，前四个都是控件变量，最后一个是自定义的MyDatabaseHelper对象用来管理数据库。



初始化如下，控件变量就是获取布局控件，MyDatabaseHelper则调用构造函数新建对象。



在onCreate方法中，除了初始化之外，只有一个监听器。为增加按钮绑定一个监听器，监听单击事件。监听到单击后，获取3个文本编辑框的内容。然后判断如果名字为空，则弹出Toast消息进行提示，如果不为空，则向数据库中插入该条数据，如果数据库中已经有该名字的记录而导致插入失败，则弹出另一条Toast消息提示。如果插入成功，则调用finish结束这个activity，返回主界面。



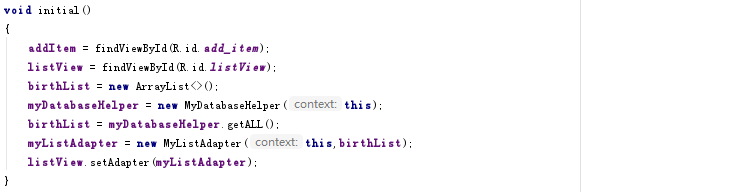
四、MainAcitivity中的相关代码。

MainActivity中内容稍多，所以我们放到最后来讲。

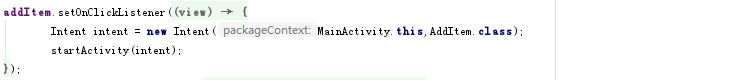
变量如下所示，有控件变量，自定义的ListView适配器MyListAdapter，一个作为ListView数据的列表birthList，还有数据库操作的对象MyDatabaseHelper。



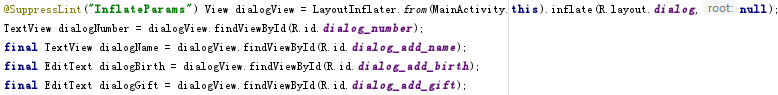
变量的初始化如下所示，控件变量就是获取布局控件，MyDatabaseHelper则调用构造函数初始化，然后调用其获取所有数据的getAll方法并将其结果存到birthList中，然后使用这个List，初始化适配器MyListAdapter。最后是为ListView设置适配器。



在onCreate函数中，主要还是绑定监听器。首先是增加条目按钮的监听器，监听到单击事件后，我们就使用intent，跳转到增加条目的AddItem 的activity。这个activity之前已经讲过了。

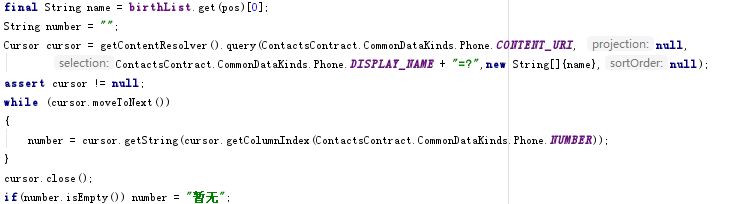


接下来是ListView中每一项的单击事件和长按事件。由于单击事件中，我们要自定义对话框，所以代码稍多，我们把这个监听器的代码分开来看。下边这段代码中，首先使用inflate方法获取我们自定义对话框的布局文件。接下来声明控件变量，获取这个布局文件中的控件，以便我们稍后填写修改信息使用。



接下来是获取通讯录中的数据。首先声明一个字符串name，赋值为列表中这一项的名字。然后就是使用getContentResolver和query方法获取通信录。getContentResolver返回一个ContentResolver对象，然后使用query进行查询，这里的查询和数据库中的query语句并不相同。这里的query有5个形参，第一个是url型，代表所有查询的数据的连接，这里填写的是通信录，这个连接是已经封装好的，我们只需正确引用即可。第二个参数是查询的列，是字符集类型，如果是null就是查询所有列。第三个形参是查询条件，也就是AQL语句中where之后的部分，科员使用替代符，第四个形参就是替换替代符的字符集，这里我们直接查询我们刚刚取到的名字所在的记录。最后一个形参是排序方式。

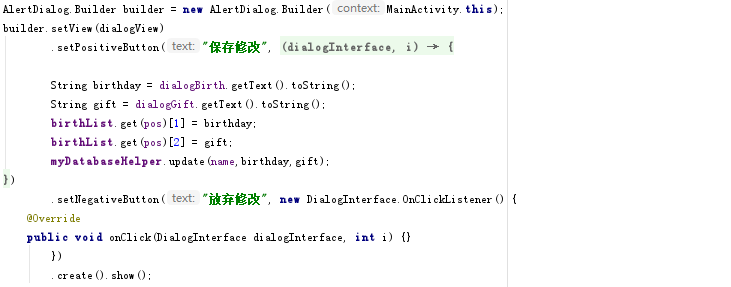
查询的结果也是一个cursor，这个对象之前数据库时已经详细讲过了，我们从中取出电话号码列即可，记得要关闭cursor。如果取出的号码为空，我们就赋值为暂无。



根据名字获取到电话号码之后，我们就可以设置自定义对话框中的内容了，一一对应设置即可。

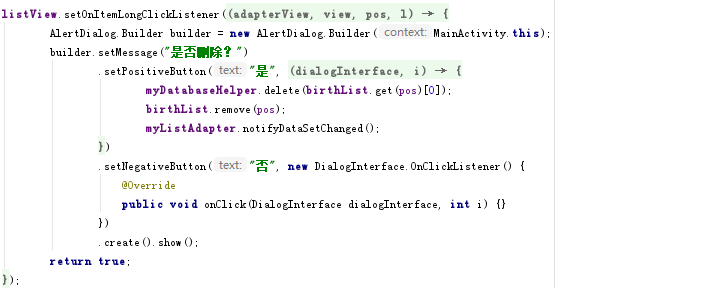


接下来我们声明一个AlertDialog.Builder对象来建立对话框。我们调用setView方法，将对话框设置成我们自定义的布局文件，这样就实现了对话框的自定义。



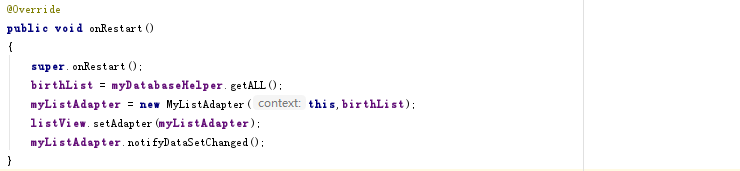
接下来就是设置对话框的确定及取消按钮及其点击事件。如果点击了放弃修改，就什么也不用做，如果点击了确认修改，我们读取对话框中文本编辑框的内容，然后调用myDatabaseHelper的update方法，根据名字修改相应的记录。对于列表的这一项，也要进行更新。最后记得调用create和show方法创建和显示对话框。

下边是ListView每一项的长按事件。监听到长按事件后，我们也是声明AlertDialog.Builder对象创建一个对话框。我们使用setMessage方法，设置需要显示的信息。然后也需要设置确认和取消按钮及其点击事件。如果点击了否，就什么也不做。如果点击了是，那么首先在数据库中根据名字删除这条数据，调用了myDatabaseHelper的delete方法。然后在列表birthList中删除这一项。还需要调用适配器的notifyDataSetChanged方法，调整显示项。最后也要使用create和show方法创建和显示对话框。



上边的就是整个onCreate方法中的代码。这一次，我们还需要重载activity的onRestart方法。这里需要了解activity的生命周期。OnCreate方法是在activity创建时调用的，onRestart方法是从当前activity调到另一个activity，另一个activity被finish回到当前activity时，所调用的方法。

放到这次实验中，我们跳转到AddItem的activity添加条目时，数据库和列表项都发生了变化。所以，在AddItem被finish之后，我们就需要在onRestart方法中，对列表和ListView进行更新。首先，我们重新获取数据库的列表，然后使用这个列表重新创建适配器，并将新适配器与ListView绑定。最后仍然需要调用notifyDataSetChanged方法调整列表项。



最后，获取通讯录时，需要静态注册读取通讯录的权限，在注册文件中加入如下一句即可。另外需注意，在手机设置中需要给与APP读取通讯录权限。



1. 实验遇到困难以及解决思路
2. 使用cursor获取通讯录时报错。

报的错是对于空对象的引用，最后解决方法就是完全照着助教给的打一遍。主要是，对cursor没有用while(cursor.toMoveNext)来获取其中数据，然后就会报错，照写之后就可以了。

2、无法正确获取通讯录权限。

起初以为通讯录权限只需要静态注册，结果报错了。但其实报错是因为上面那个问题。后来添加了动态权限，也不行，当然也是因为上边那个问题。上边那个问题解决后，发现只要静态的还是不行，但是并不需要去动态获取，只要在手机相关设置中给与APP读取通讯录权限即可。

3、使用query语句获取通讯录时，形参报错。

刚开始以为，ContentResolver的query语句和SQLite是一样的，结果错了，后来一查才知道原来形参是不一样的，照着网上的形参说明，一个个填好，最后就可以了。然后也没完全按照助教给的方式写。

# 课后实验结果

都是课后做的。

## 实验思考及感想

在这次的实验中，我们学习使用了android中内置的轻量数据库SQLite存储APP的数据，以及使用ContentProvider获取共享数据（通讯录）。由于在我们的其中Project中，我们便使用了数据库，并且那部分还是我写的，所以数据库这里对我而言不是很难。当然这次和上次期中写的还是有些不一样的。上次还额外写了一个DataBaseManager的类来管理数据库，这里把全部方法写进那个自定义的SQLiteOpenHelper类中去了。在获取通信录方面，就是权限和那个query的用法比较难弄。另外自定义对话框结合课件和网上的内容，很快就搞定了。

作业要求：

1. 命名要求: 学号\_姓名\_实验编号，例如15330000\_林XX\_lab1。
2. 实验报告提交格式为pdf。
3. 实验内容不允许抄袭，我们要进行代码相似度对比。如发现抄袭，按0分处理。