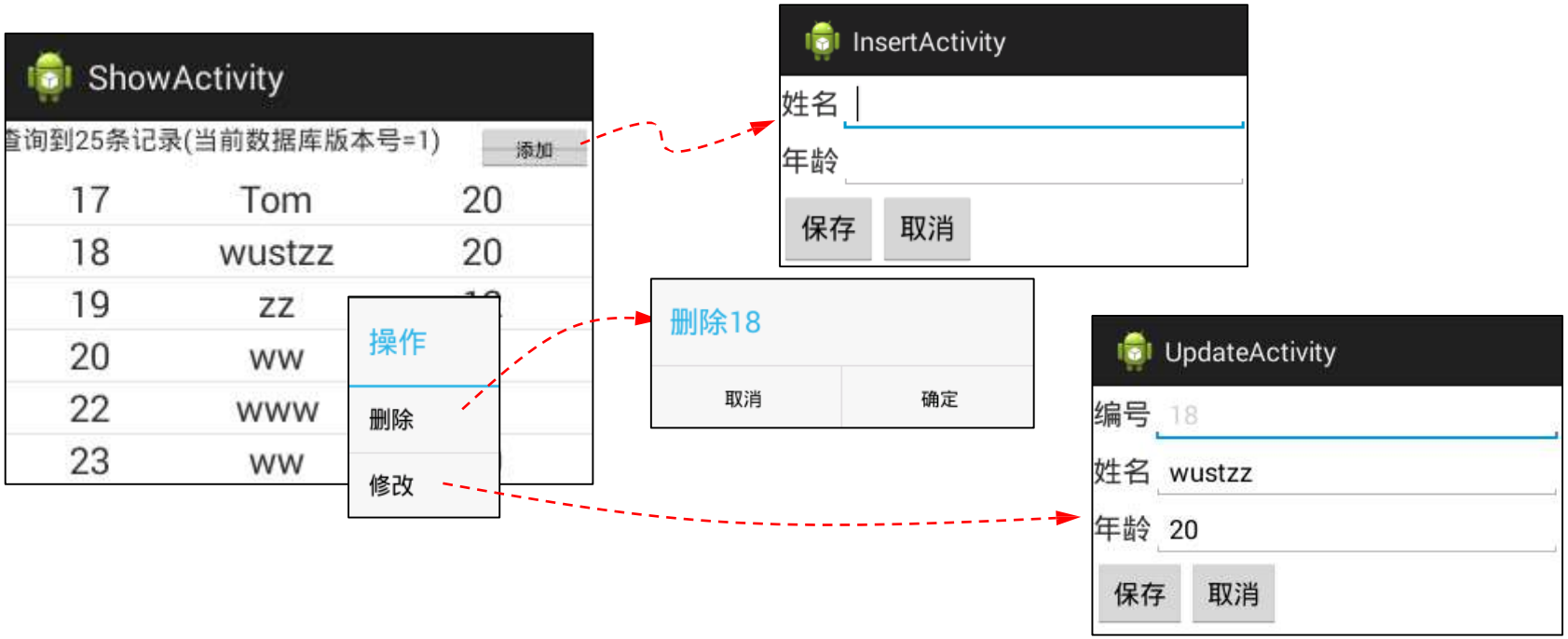
Android应用开发

主讲教师 胡鑫

hithuxin@hit.edu.cn

主要内容

1. ListView显示数据库数据（ShowActivity）
2. 添加记录功能（InsertActivity）
3. 给ListView添加ContextMenu菜单：删除+编辑（UpdateActivity）



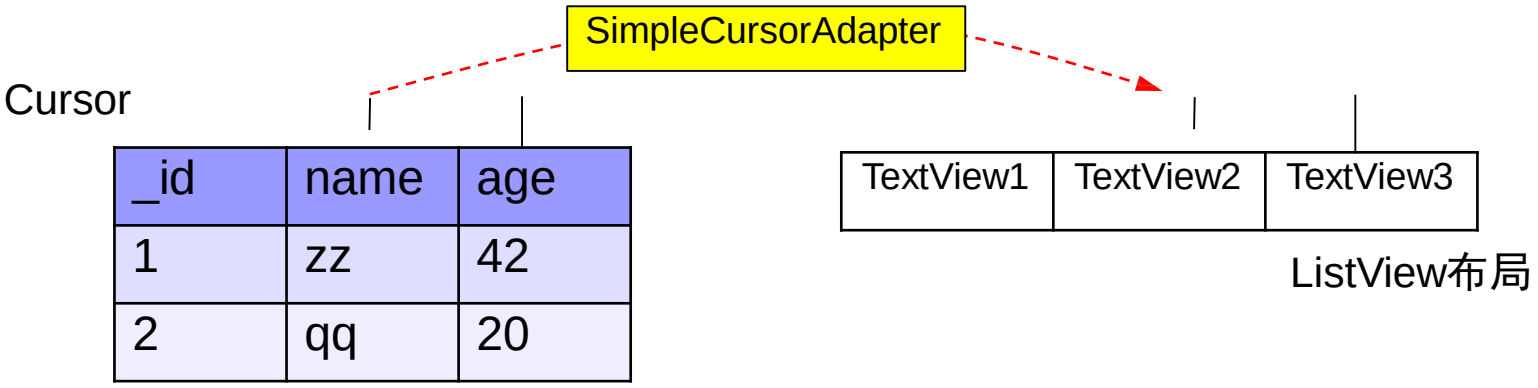
1. ListView显示数据库数据（ShowActivity）

ListView 2个关键点：

（1）自定义layout布局；

（2）使用SimpleCursorAdapter适配器填充：(用于显示数据库数据)

该适配器允许将一个Cursor(查询结果集)的数据列绑定到ListView自定义布局中的TextView 或 ImageView组件上(详见后面代码)



（1）ListView自定义layout布局

- 新建一个名为listview.xml布局：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="horizontal" >
  <TextView android:id="@+id/txtID"
    android:layout_width="100dip"
    android:layout_height="wrap_content"
    android:text="id"
    android:textSize="20dp"
    android:gravity="center"/>
  <TextView
    android:id="@+id/txtName"
    android:layout_width="100dip"
    android:layout_height="wrap_content"
    android:text=" name"
    android:textSize="20dp"
    android:gravity="center"/>
  <TextView android:id="@+id/txtAge"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
```

（2）使用SimpleCursorAdapter填充ListView

- SimpleCursorAdapter基本方法：

```
String[ ] from = { "_id", "name", "age" };
int[ ] to = { R.id.txtID, R.id.txtName, R.id.txtAge };
SimpleCursorAdapter adapter =
new SimpleCursorAdapter(this, R.layout.listview, cursor, from, to);
```

要显示的Cursor的列

显示数据的组件id

上下文

自定义的ListView布局

select查询返回的cursor

特别注意： SimpleCursorAdapter要求返回的结果集cursor中必须包含_id字段，所以需要 对取得结果集进行处理特殊处理一下（见后）

- ShowAcitvity的OnCreate()主要代码：

```
DBHelper helper = new DBHelper(getApplicationContext(), "test.db", null,1);
SQLiteDatabase db=helper.getWritableDatabase();
Cursor cursor = db.rawQuery( "SELECT id as _id, name,age
                                                    FROM person",null);

String[] from = { "_id", "name", "age" };
int[] to = { R.id.txtID, R.id.txtName, R.id.txtAge };
SimpleCursorAdapter adapter = new SimpleCursorAdapter(
                                                    this, R.layout.listview, cursor, from, to);

//cursor.close();
ListView li=(ListView)findViewById(R.id.listView1);

li.setAdapter(adapter);
TextView tv=(TextView)findViewById(R.id.textView1);
tv.setText("查询到"+cursor.getCount()+"条记录");
```

Cursor使用_id字段

不能close()，否则SimpleCursorAdapter将不能从Cursor中读取数据显示

2. 添加记录（InsertActivity）

关键点：两个Activity之间的数据传递 ShowActivity打开InsertActivity，后者完成添加记录，并回传 一个值让ShowActivity刷新以显示新添加的记录。

ShowActivity“添加”按钮

```
Button bt1= (Button)findViewById(R.id.button1);  //注意id值
bt1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View arg0) {
        Intent intent=new Intent(ShowActivity.this,InsertActivity.class);
        startActivityForResult(intent, 100);
    }
});
```

requestCode设置为100
在后面的onActivityResult()中会用到

InsertActivity“保存”按钮

```
Button bt1= (Button)findViewById(R.id.newRec);
bt1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View arg0) {
        EditText   et1=(EditText)findViewById(R.id.editText1);
        EditText   et2=(EditText)findViewById(R.id.editText2);
        Person person=new Person(et1.getText().toString(),
                                 Integer.parseInt(et2.getText().toString()));
        insert(person);  //代码后页
        setResult(RESULT_OK, null);
        finish();
    }
});
```

不回传intent
可设为null

其中insert()方法

```
public void insert(Person person){
    DBHelper helper = new DBHelper(getApplicationContext(), "test.db", null,1);
    SQLiteDatabase db=helper.getWritableDatabase();
    String sql="INSERT INTO person VALUES (NULL, ?, ?)";
    db.execSQL(sql, new Object[ ] { person.name, person.age } );
    db.close();
    Toast.makeText(getApplicationContext(), "记录添加成功",
    Toast.LENGTH_SHORT).show();
}
```

ShowActivity接收回传信息（下面代码与onCreate并列）

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode==100)
        if(resultCode==RESULT_OK){
         onCreate(null);
    }
}
```

100表明是来自于InsertActivity的回传

ShowActivity重新执行onCreate，即完成刷新

3. ListView添加ContextMenu (ShowActivity)

- 主要步骤：

第1步：新建菜单资源

（1）在res下新建menu目录

（2）在menu目录下新建一个manage.xml文件，代码：

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/delete"  android:title="删除"></item>
    <item android:id="@+id/update"  android:title="修改"></item>
</menu>
```

第2步：创建上下文菜单

（下面代码放在ShowActivity中，与onCreate并列）

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
                                ContextMenuInfo menuInfo) {
        menu.setHeaderTitle("操作");
        getMenuInflater().inflate(R.menu.manage, menu);
}
```

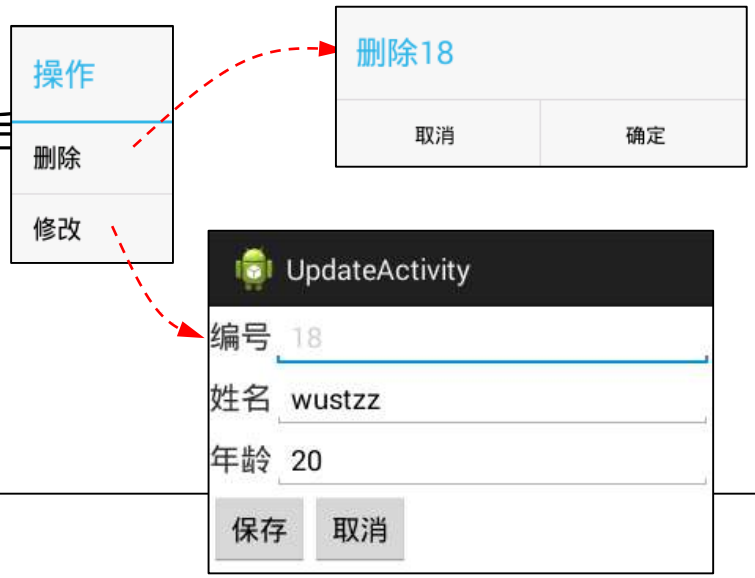
上下文菜单弹出时的标题栏

菜单资源文件

第3步：添加上下文菜单选中项方法

（下面代码放在ShowActivity中，与onCreateContextMenu并列）

```
@Override
public boolean onContextItemSelected(MenuItem item) {
    switch ( item.getItemId() ) {
        case R.id.delete:
            delete(item);    //代码见后
            return true;
        case R.id.update :
            update(item);    //代码见后
            return true;
        default:
            return false;
    }
}
```



第4步：将上下文菜单注册到ListView上（完）

（下面代码放在ShowActivity中onCreate中）

```
registerForContextMenu( li );
```

```
ListView li=(ListView)findViewById(R.id.listView1);
```

后续删除、修改操作分析：

- 两个操作的关键点：
  - 如何得到选中行的id值以及其他字段值（name、age）。
  - 删除：根据id值作为删除记录的条件。
  - 修改：将id值及其他字段的值传到UpdateActivity显示和修改。

关键点代码：

在上下文菜单选中项onContextItemSelected(MenuItem item)方法中：

- 使用item.getMenuInfo()可获得选中行上下文菜单信息：

```
AdapterView.AdapterContextMenuInfo info =
        (AdapterView.AdapterContextMenuInfo)item.getMenuInfo();
```

- 然后使用上面的info对象：
  - info.id：可获得记录id值
  - info.targetView：可获得选中行所有对象的Views集合

然后可以使用

(TextView)info.targetView.findViewById(R.id.××)：
可得到当前选中行中的某个TextView元素

具体代码：删除操作 delete(MenuItem item)方法

```
public void delete(MenuItem item){
    final AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo)item.getMenuInfo();
    if(info.id>0){
            new AlertDialog.Builder(this).setTitle("删除" + info.id)
            .setPositiveButton("确定", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int which) {
                    DBHelper helper = new DBHelper(getApplicationContext(), "test.db", null,1);
                    SQLiteDatabase db=helper.getWritableDatabase();
                    db.execSQL( "Delete from person where id=?", new Object[ ]{ info.id } );
                    db.close();
                    Toast.makeText(getApplicationContext(), "记录删除成功", Toast.LENGTH_SHORT).show();
                    onCreate(null);     //重新加载一次Activity，刷新
                }
            })
            .setNegativeButton("取消", null) .show();
    }
}
```

（2）修改操作

将当前选中行的id、name、age字段值送到UpdateActivity中去修改，后者回传信息

```
public void update(MenuItem item){
    final AdapterView.AdapterContextMenuInfo info= (AdapterView.AdapterContextMenuInfo)item.getMenuInfo();
    Intent intent =new Intent(ShowActivity.this, UpdateActivity.class);
    Bundle bundle=new Bundle();
     bundle.putString("id",
    String.valueOf(info.id));
    bundle.putString("username",((TextView)info.targetView.findViewById(R.id.txtName)).getText().toString());
    bundle.putString("age",((TextView)info.targetView.findViewById(R.id.txtAge)).getText().toString());
    intent.putExtras(bundle);
    startActivityForResult(intent, 200);
}
```

requestCode设置为200
在后面的onActivityResult()中会用到

UpdateActivity：oncreat()代码

显示ShowActivity传来的

id、name、age字段值

```
Intent intent=getIntent();
Bundle bundle=intent.getExtras();
String id=bundle.getString("id");
String name=bundle.getString("username");
String age=bundle.getString("age");
final EditText et1=(EditText)findViewById(R.id.editText1);
final EditText et2=(EditText)findViewById(R.id.editText2);
final EditText et3=(EditText)findViewById(R.id.editText3);
et1.setText(id);
et2.setText(name);
et3.setText(age);
```

UpdateActivity(代码续)

```
Button bt1= (Button)findViewById(R.id.button1);
bt1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View arg0) {
        DBHelper helper = new DBHelper(getApplicationContext(), "test.db", null,1);
        SQLiteDatabase db=helper.getWritableDatabase();
        //更新数据，id值不能修改
        db.execSQL("Update person set name=? , age=? where id=?", new Object[ ]{
                                et2.getText().toString(),et3.getText().toString(),et1.getText().toString() } );
        db.close();
        Toast.makeText(getApplicationContext(), "记录修改成功", Toast.LENGTH_SHORT).show();
        setResult(RESULT_OK, null);
        finish();  //不能少
  }
});
```

UpdateActivity(代码续)

```
Button bt2= (Button)findViewById(R.id.button2);
bt2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View arg0) {
            finish();
        }
});
```

取消按钮

ShowActivity接收回传信息

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    ….
    if(requestCode==200)
    if(resultCode==RESU
    LT_OK){
        onCreate(null);
    }
}
```

200表明是来自于InsertActivity的回传

ShowActivity重新执行onCreate，即完成刷新

【完】