**《移动应用开发》**

**实验报告**

**班级：软件161**

**姓名：胡启明**

**学号：164547**

**实验四 Android生命周期**

**一、实验目的**

1. 加深对用户界面的控件的理解和使用。

2．掌握各种界面布局的用法。

3．掌握选项菜单、子菜单和快捷菜单的使用方法。

**二、实验内容**

1. 建立一个“ButtonDemo”的程序，包含Button和ImageButton两个按钮，上方是“Button按钮”，下方是一个ImageButton控件？

2. 建立一个“CheckboxRadiobuttonDemo”程序，包含五个控件，从上至下分别是TextView01、CheckBox01、 CheckBox02、RadioButton01、RadioButton02，当选择RadioButton01， RadioButton02则无法选择？

3．建立一个“TabDemo”程序，包含三个XML文件，分别为tab1.xml、tab2.xml和tab3.xml，这3个文件分别使用线性布局、相对布局和绝对布局示例中的main.xml的代码，并将布局的ID分别定义为layout01、layout02和layout03？

4．建立一个“MyContextMenu”程序，来完成Android菜单的功能。

**三、实验仪器、设备**

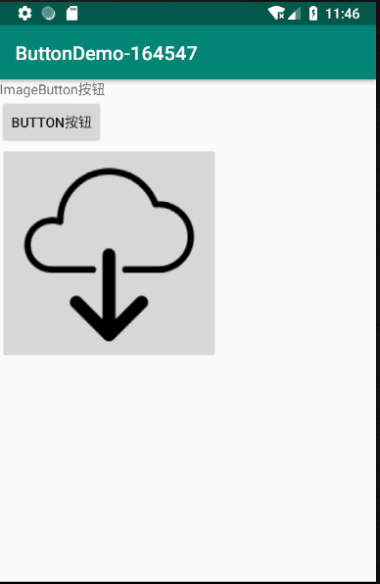
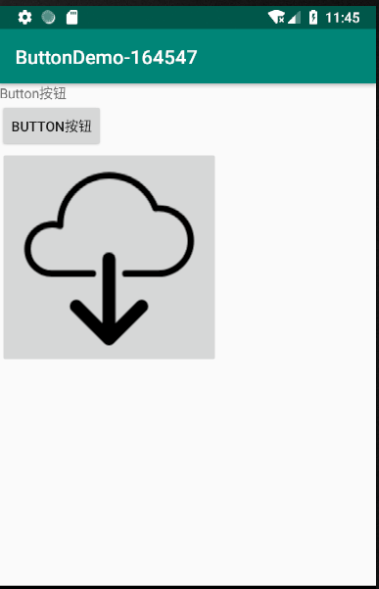
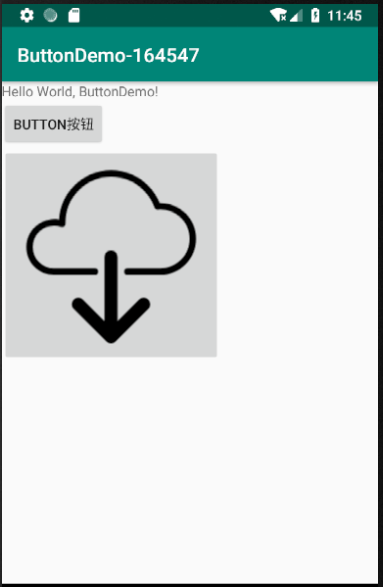
硬件：PC 微型计算机、1G以上内存，40G以上硬盘

软件：Windows XP，Eclipse , JDK , Android SDK

**四、实验步骤**

1.ButtonDemo

1). 以下分别为初始状态、点击Button按钮和点击Imagebutton后的结果。



2). 代码实现如下：

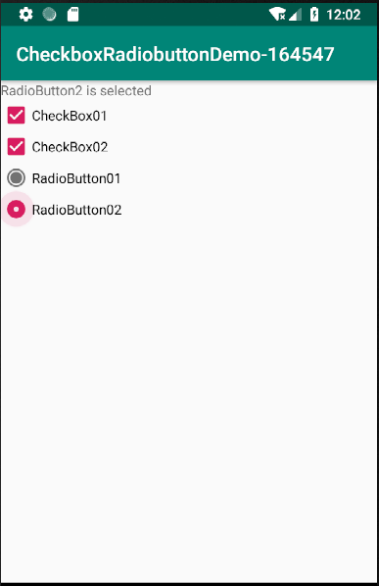
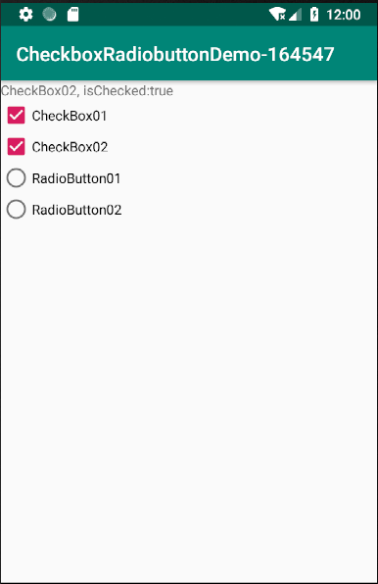
ButtonDemo.java  
*/\*\*  
 \* 对于实现点击事件监听上，还有一种方法  
 \* 就是通过实现接口的方式  
 \* 然后重写接口的onClick方法，最后将组件绑定到当前类的对象上(通过this引用的方式)  
 \*/***public class** ButtonDemo **extends** AppCompatActivity **implements** View.OnClickListener {  
  
 **private** TextView **textView**;  
 **private** Button **button**;  
 **private** ImageButton **imageButton**;  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 initViews();  
 }  
  
 **public void** initViews() {  
 **textView** = findViewById(R.id.***TextView01***);  
 **button** = findViewById(R.id.***Button01***);  
 **imageButton** = findViewById(R.id.***ImageButton01***);  
  
 **button**.setText(**"Button按钮"**);  
 **imageButton**.setImageResource(R.drawable.***download***);  
  
 **button**.setOnClickListener(**this**);  
 **imageButton**.setOnClickListener(**this**);  
 }  
  
 @Override  
 **public void** onClick(View v) {  
 **switch** (v.getId()) {  
 **case** R.id.***Button01***:  
 **textView**.setText(**"Button按钮"**);  
 **break**;  
 **case** R.id.***ImageButton01***:  
 **textView**.setText(**"ImageButton按钮"**);  
 **break**;  
 }  
 }  
}

activity\_main.xml

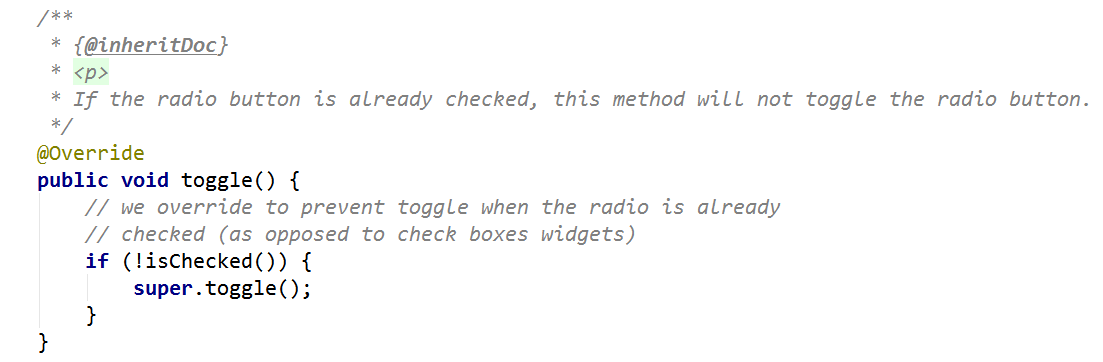
<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"**>  
 <**TextView android:id="@+id/TextView01"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/hello"**/>  
 <**Button android:id="@+id/Button01"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Button01"** >  
 </**Button**>  
  
 <**ImageButton android:id="@+id/ImageButton01"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"**>  
 </**ImageButton**>  
</**LinearLayout**>

2.CheckboxRadiobuttonDemo

1). 在本题中可以发现，两个CheckBox是可以同时勾选的，但是同一组的RadioButton是不允许同时选择的。可以看到，在选择RadioButton02的瞬间，RadioButton01的选择即被取消。



同时通过查看RadioButton的源码可以发现，之所以无法通过再次点击取消已选择的单选项是因为源码中的如下判断操作。



2). 代码实现如下：

CheckboxRadiobuttonDemo.java

**package** com.example.checkboxradiobuttondemo;  
  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.CheckBox;  
**import** android.widget.RadioButton;  
**import** android.widget.TextView;  
  
*/\*\*  
 \* 和上题类似，对代码风格进行了简单重构  
 \*/***public class** CheckboxRadiobuttonDemo **extends** AppCompatActivity **implements** View.OnClickListener {  
 **private** CheckBox **checkBox1**;  
 **private** CheckBox **checkBox2**;  
 **private** RadioButton **radioButton1**;  
 **private** RadioButton **radioButton2**;  
 **private** TextView **textView**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 initViews();  
 }  
  
 **private void** initViews() {  
 **textView** = findViewById(R.id.***TextView01***);  
 **checkBox1** = findViewById(R.id.***CheckBox01***);  
 **checkBox2** = findViewById(R.id.***CheckBox02***);  
 **radioButton1** = findViewById(R.id.***RadioButton01***);  
 **radioButton2** = findViewById(R.id.***RadioButton02***);  
  
 **checkBox1**.setOnClickListener(**this**);  
 **checkBox2**.setOnClickListener(**this**);  
 **radioButton1**.setOnClickListener(**this**);  
 **radioButton2**.setOnClickListener(**this**);  
 }  
  
 @Override  
 **public void** onClick(View v) {  
 **switch** (v.getId()) {  
 **case** R.id.***CheckBox01***:  
 **textView**.setText(**"CheckBox01, isChecked:"** + String.*valueOf*(**checkBox1**.isChecked()));  
 **break**;  
 **case** R.id.***CheckBox02***:  
 **textView**.setText(**"CheckBox02, isChecked:"** + String.*valueOf*(**checkBox2**.isChecked()));  
 **break**;  
 **case** R.id.***RadioButton01***:  
 **textView**.setText(**"RadioButton1 is selected"**);  
 **break**;  
 **case** R.id.***RadioButton02***:  
 **textView**.setText(**"RadioButton2 is selected"**);  
 **break**;  
 }  
  
 }  
}

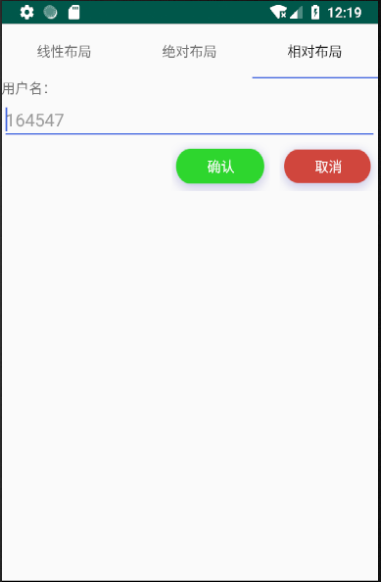
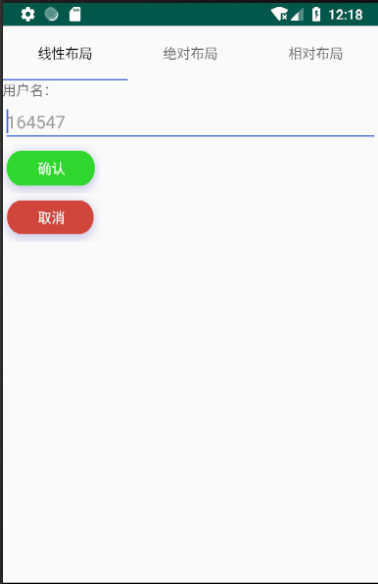
activity\_main.xml

<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 android:orientation="vertical"**>  
  
 <**TextView  
 android:id="@+id/TextView01"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/hello"** />  
  
 <**CheckBox  
 android:id="@+id/CheckBox01"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="CheckBox01"** />  
  
 <**CheckBox  
 android:id="@+id/CheckBox02"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="CheckBox02"** />  
  
 <**RadioGroup  
 android:id="@+id/RadioGroup01"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"**>  
  
 <**RadioButton  
 android:id="@+id/RadioButton01"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="RadioButton01"** />  
  
 <**RadioButton  
 android:id="@+id/RadioButton02"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="RadioButton02"** />  
 </**RadioGroup**>  
</**LinearLayout**>

3.TabDemo

1). 如下所示为三个不同的Tab页面

在这里将组件的基础颜色修改为蓝色，并为按钮设置了Background属性，修改了一下基本样式。



2). 代码实现如下：

TabDemo.java

**package** com.example.tabhost;  
  
**import** android.app.TabActivity;  
**import** android.os.Bundle;  
**import** android.view.LayoutInflater;  
**import** android.widget.TabHost;  
  
**public class** MainActivity **extends** TabActivity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
  
 TabHost tabHost = getTabHost();  
  
 LayoutInflater.*from*(**this**).inflate(R.layout.***tab1***, tabHost.getTabContentView(),**true**);  
 LayoutInflater.*from*(**this**).inflate(R.layout.***tab2***, tabHost.getTabContentView(),**true**);  
 LayoutInflater.*from*(**this**).inflate(R.layout.***tab3***, tabHost.getTabContentView(),**true**);  
  
 tabHost.addTab(tabHost.newTabSpec(**"TAB1"**).  
 setIndicator(**"线性布局"**).setContent(R.id.***layout01***));  
 tabHost.addTab(tabHost.newTabSpec(**"TAB2"**).  
 setIndicator(**"绝对布局"**).setContent(R.id.***layout02***));  
 tabHost.addTab(tabHost.newTabSpec(**"TAB3"**).  
 setIndicator(**"相对布局"**).setContent(R.id.***layout03***));  
  
 }  
}

tab1.xml

<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/layout01"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"**>  
  
 <**TextView  
 android:id="@+id/label"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="用户名："** />  
  
 <**EditText  
 android:id="@+id/entry"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="164547"** />  
  
 <**Button  
 android:id="@+id/ok"  
 android:layout\_width="100dp"  
 android:layout\_height="50dp"  
 android:background="@drawable/btn\_green"  
 android:text="确认"  
 android:textColor="#FFF"** />  
  
 <**Button  
 android:id="@+id/cancel"  
 android:layout\_width="100dp"  
 android:layout\_height="50dp"  
 android:background="@drawable/btn\_red"  
 android:text="取消"  
 android:textColor="#FFF"** />  
</**LinearLayout**>

tab2.xml

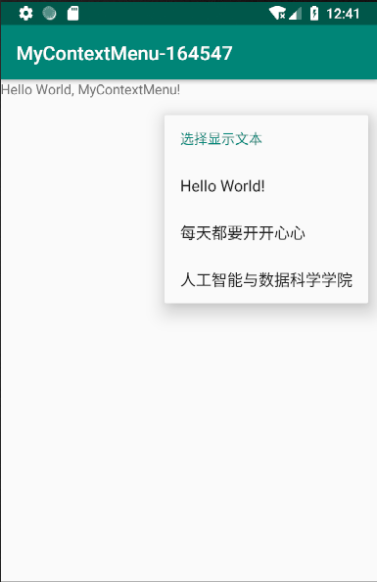
<**AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/layout02"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"**>  
  
 <**TextView  
 android:id="@+id/label"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_x="40dip"  
 android:layout\_y="40dip"  
 android:text="用户名："** />  
  
 <**EditText  
 android:id="@+id/entry"  
 android:layout\_width="150dip"  
 android:layout\_height="wrap\_content"  
 android:layout\_x="40dip"  
 android:layout\_y="60dip"  
 android:hint="164547"** />  
  
 <**Button  
 android:id="@+id/ok"  
 android:layout\_width="100dp"  
 android:layout\_height="50dp"  
 android:layout\_x="40dip"  
 android:layout\_y="120dip"  
 android:background="@drawable/btn\_green"  
 android:text="确认"  
 android:textColor="#FFF"** />  
  
 <**Button  
 android:id="@+id/cancel"  
 android:layout\_width="100dp"  
 android:layout\_height="50dp"  
 android:layout\_x="150dip"  
 android:layout\_y="120dip"  
 android:background="@drawable/btn\_red"  
 android:text="取消"  
 android:textColor="#FFF"** />  
</**AbsoluteLayout**>

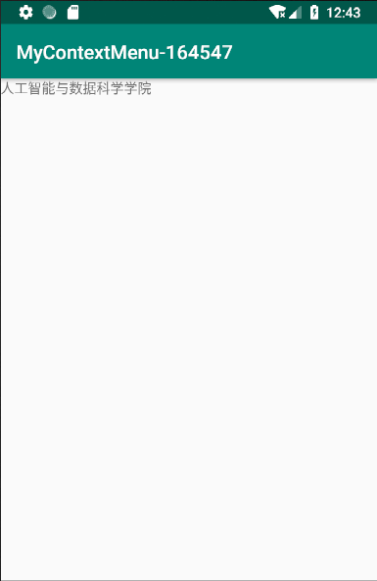
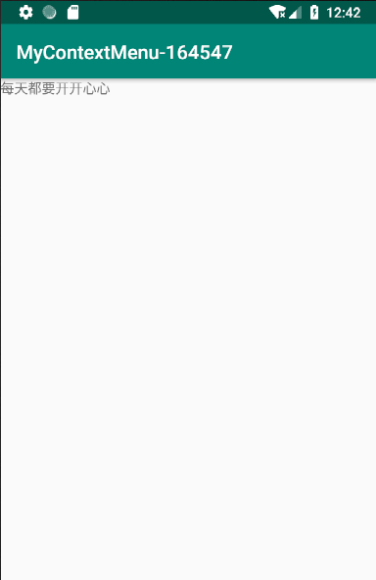
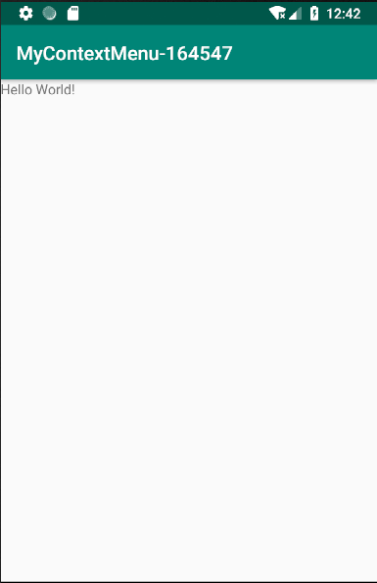
tab3.xml

<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/layout03"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"**>  
  
 <**TextView  
 android:id="@+id/label"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="用户名："** />  
  
 <**EditText  
 android:id="@+id/entry"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/label"  
 android:hint="164547"** />  
  
  
 <**Button  
 android:id="@+id/ok"  
 android:layout\_width="100dp"  
 android:layout\_height="50dp"  
 android:layout\_alignTop="@id/cancel"  
 android:layout\_toLeftOf="@id/cancel"  
 android:background="@drawable/btn\_green"  
 android:text="确认"  
 android:textColor="#FFF"** />  
  
 <**Button  
 android:id="@+id/cancel"  
 android:layout\_width="100dp"  
 android:layout\_height="50dp"  
 android:layout\_below="@id/entry"  
 android:layout\_alignParentRight="true"  
 android:layout\_marginLeft="10dip"  
 android:background="@drawable/btn\_red"  
 android:text="取消"  
 android:textColor="#FFF"** />  
</**RelativeLayout**>

4.MyContextMenu

1). 如下所示为分别为菜单和点击菜单的不同选项后TextView的变化情况。





2). 代码实现如下：

**package** com.example.mycontextmenu;  
  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.ContextMenu;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.TextView;  
  
*/\*\*  
 \* 本实验中封装了原本代码中的重复操作  
 \* 并使用数组的方式来管理题目中需要用到的常量  
 \* 这样可以通过循环和简单的计算来实现原本代码中的重复操作，提升了代码的可拓展性  
 \*/***public class** MainActivity **extends** AppCompatActivity {  
  
 *// 通过数组的方式声明这些常量* **final static int**[] ***CONTEXT\_MENUS*** = **new int**[]{  
 Menu.***FIRST***,  
 Menu.***FIRST*** + 1,  
 Menu.***FIRST*** + 2  
 };  
  
 **final static** String[] ***STRINGS*** = **new** String[]{  
 **"Hello World!"**,  
 **"每天都要开开心心"**,  
 **"人工智能与数据科学学院"** };  
  
  
 TextView **LabelView** = **null**;  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **LabelView** = findViewById(R.id.***label***);  
 registerForContextMenu(**LabelView**);  
 }  
  
 @Override  
 **public void** onCreateContextMenu(ContextMenu menu,  
 View v, ContextMenu.ContextMenuInfo menuInfo) {  
 menu.setHeaderTitle(**"选择显示文本"**);  
  
 **for** (**int** i = 0; i < ***CONTEXT\_MENUS***.**length**; i++) {  
 menu.add(0, ***CONTEXT\_MENUS***[i], 0, ***STRINGS***[i]);  
 }  
  
 }  
  
 @Override  
 **public boolean** onContextItemSelected(MenuItem item) {  
  
 *// 通过检查异常的方式确定返回值为true还是false* **try**{  
 **LabelView**.setText(***STRINGS***[item.getItemId()-***CONTEXT\_MENUS***[0]]);  
 **return true**;  
 }**catch** (IndexOutOfBoundsException e){  
 **return false**;  
 }  
 }  
  
}

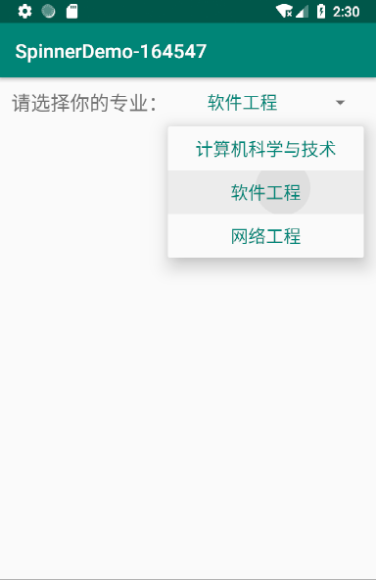
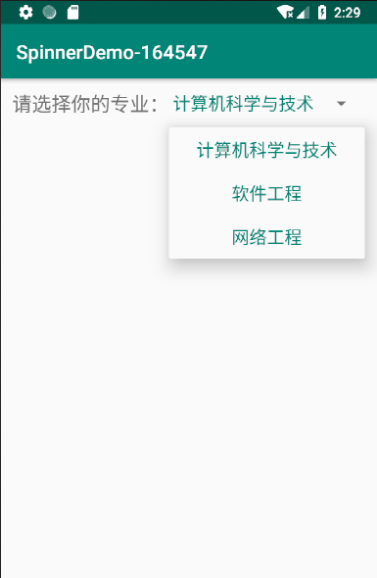
activity\_main.xml

<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 android:orientation="vertical"**>  
  
 <**TextView  
 android:id="@+id/label"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 android:text="@string/hello"** />  
</**LinearLayout**>

**五、实验思考题**

1.建立一个程序“SpinnerDemo”包含3个子项Spinner控件？

1). 功能实现如下



2). 代码实现如下

MainActivity.java

**package** com.hebut.spinnerdemo;  
  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.Spinner;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 **private** Spinner **mSpinnerSimple**;  
 **private final static** String[] ***spinnerItems*** = {**"计算机科学与技术"**, **"软件工程"**, **"网络工程"**};  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 initSpinner();  
 }  
  
 */\*\*  
 \* Spinner自定义样式  
 \* 1、Spinner内的TextView样式：item\_select  
 \* 2、Spinner下拉中每个item的TextView样式：item\_drop  
 \* 3、Spinner下拉框样式，属性设置  
 \*/* **public void** initSpinner() {  
 **mSpinnerSimple** = findViewById(R.id.***spinner\_simple***);  
  
 **mSpinnerSimple**.setDropDownWidth(400);  
 **mSpinnerSimple**.setDropDownHorizontalOffset(100);  
 **mSpinnerSimple**.setDropDownVerticalOffset(100);  
  
 ArrayAdapter<String> spinnerAdapter = **new** ArrayAdapter<>(**this**, R.layout.***item\_select***, ***spinnerItems***);  
 *//自定义下拉的字体样式* spinnerAdapter.setDropDownViewResource(R.layout.***item\_drop***);  
 **mSpinnerSimple**.setAdapter(spinnerAdapter);  
 }  
}

activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:orientation="horizontal"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="50dp"  
 android:gravity="center"  
 android:text="请选择你的专业："  
 android:textSize="20sp"** />  
  
 <**Spinner  
 android:id="@+id/spinner\_simple"  
 android:layout\_width="200dp"  
 android:layout\_height="match\_parent"** />  
</**LinearLayout**>

item\_drop.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**TextView  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="10dp"  
 android:gravity="center"  
 android:textColor="@color/colorPrimary"  
 android:textSize="18sp"**/>

item\_select.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**TextView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:textColor="@color/colorPrimary"  
 android:textSize="18sp"** />

2．建立一个“ListViewDemo”程序，包含四个控件，从上至下分别为TextView01、ListView01、 ListView02和 ListView03？

1). 功能实现如下



2). 代码实现如下

MainActivity.java

**package** com.hebut.listviewdemo;  
  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.ListAdapter;  
**import** android.widget.ListView;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 **private** ListView **ListView01**,**ListView02**,**ListView03**;  
  
 **private** String[][] **data** = {  
 {**"苏小娜"**,**"周鹏镯"**,**"王浩然"**,**"赵琦"**,**"喻岚"**},  
 {**"张雨晴"**,**"刘蕊"**,**"史清杰"**,**"马思雨"**,**"周铭瑞"**},  
 {**"罗柳芳"**,**"褚丽爽"**,**"郑佳奇"**,**"王红梅"**,**"姚楚琳"**}  
 };  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 initViews();  
 }  
  
 **private void** initViews() {  
 **ListView01** = findViewById(R.id.***list\_view\_01***);  
 **ListView02** = findViewById(R.id.***list\_view\_02***);  
 **ListView03** = findViewById(R.id.***list\_view\_03***);  
  
 **ListView01**.setAdapter(**new** ArrayAdapter<>(**this**,R.layout.***simple\_list\_item***,  
 R.id.***item\_text***,**data**[0]));  
 **ListView02**.setAdapter(**new** ArrayAdapter<>(**this**,R.layout.***simple\_list\_item***,  
 R.id.***item\_text***,**data**[1]));  
 **ListView03**.setAdapter(**new** ArrayAdapter<>(**this**,R.layout.***simple\_list\_item***,  
 R.id.***item\_text***,**data**[2]));  
 }  
}

activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"**>  
  
 <**TextView  
 android:id="@+id/text\_view\_01"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:text="软件161、162、163班人员名录"  
 android:textSize="20sp"  
 android:layout\_marginBottom="30dp"**/>  
  
 <**ListView  
 android:id="@+id/list\_view\_01"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="200dp"** />  
  
 <**ListView  
 android:id="@+id/list\_view\_02"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="200dp"** />  
  
 <**ListView  
 android:id="@+id/list\_view\_03"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="200dp"** />  
  
  
 </**LinearLayout**>  
</**ScrollView**>

simple\_list\_item.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:gravity="center"**>  
  
 <**TextView  
 android:id="@+id/item\_text"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:textSize="24sp"  
 android:layout\_marginLeft="50dp"  
 android:layout\_marginStart="50dp"** />  
  
 <**CheckBox  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"**/>  
</**LinearLayout**>

3．Android界面框架是如何实现按键事件和触摸事件的？

1）按键事件通过实现View类的OnKeyListener接口，并重写onKey方法，最后使用对应控件的setOnKeyListener方法将接口绑定到控件上实现对控件按键事件的监听捕获。

其基本形式如下

View.OnKeyListener() {  
 @Override  
 **public boolean** onKey(View v, **int** keyCode, KeyEvent event) {  
 **return false**;  
 }  
};

v是接收按键事件的控件对象，keyCode是按下的物理按键的键值，event是包含了按键事件完整信息的事件对象。包括获取动作，是否长按、是否上档等方法判断。返回值用于判断事件是否被消费，如果未被消费则事件还可以沿视图树继续向下传递。

要知道keyEvent的事件分发流程是这样的：

*KeyEvent到达Activity后的传递流程。*

*1、一个完整的KeyEvent由onKeyDown和onKeyUp组成，KeyEvent首先由进入Activity的dispatchKeyEvent进行onKeyDown事件的分发。*

*2、从最外层ViewGroup进行遍历，直到找到真正拥有焦点的View并回调此事件*。

通过查找资料可以知道安卓的按键事件监听主要应用在交互设备，如遥控器手柄等，可以说Android TV的操控就全靠它了。

2）同样的，触摸事件也是通过监听器接口来实现的。

View.OnTouchListener() {  
 @Override  
 **public boolean** onTouch(View v, MotionEvent event) {  
 **return false**;  
 }  
}

与上面不同的是这里的event是MotionEvent对象的类，主要用识别一些手势，比如按下、拖动、抬起、滑动等。

**六、实验感想**

1. 好的代码风格，良好的封装有利于代码的重复利用和功能拓展，平时要养成重构代码的好习惯；

2. 现在接触到的东西往浅看很浅，而往深看则很深，但只要花时间去弄清楚其原理，剩余的高层建筑就只是表层实现的区别了。