桂林电子科技大学

序号：

课号：

**Android应用开发**  实验报告

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | 实验二 Activity、控件及布局 | | | | | | | |  | 辅导员意见：  成绩 辅导员  签 名 |
| 院 系 | 计算机与信息安全学院 | | | 专业 | | 计算机科学与技术 | | |
| 学 号 | 1900301113 | | | 姓名 | | 龚俊源 | | |
| 实验日期 | 2022 | 年 | 4 | | 月 | | 16 | 日 |
|  |  | | | | | | | |

|  |
| --- |
| **（第四小节）**  **一、实验目的**   1. 学会AndroidMainfest.xml文件中具体代码中结构 2. 学会如何使用区分Intent显式启动和隐式启动，使用显示启动Activity的方式 3. 学会为Activity添加返回父Activity的方法 |
| **二、实验内容**  代码分析：  1、activity\_main.xml：  app:layout\_constraintEnd\_toEndOf="@+id/send\_message"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"/>  指明控件属性的约束关系，指明了控件EditText的左侧和右侧和父容器ConstraintLayout绑定对齐。注意：在约束布局中，一个控件至少要指明水平方向和垂直方向上的约束关系。有了约束关系，控件的位置也确定了。  app:layout\_constraintBaseline\_toBaselineOf="@+id/message"  指明在Button控件中的文字部分的底部，这个底部要和EditText的文字底部所对齐，这样子会让我们在整个页面中看得更清晰整齐，但是baseline是不可见的，约定文字在这条线之上，好处是设置两个控件的文字保持高度一致。  可以使用app:layout\_constraintTop\_toTopOf=”@id/etMessage”和  app:layout\_constraintBottom\_toBottom来代替上面的语句，使得按钮中的字体和EditText中的字体对齐。  2、MainActivity.java  进行控件的绑定，在onCreate方法中绑定事件  String message = etMessage.getText().toString();  Intent intent = new Intent(MainActivity.this,MessageActivity.class);  intent.putExtra(MESSAGE\_STRING,message);  startActivity(intent);  Intent，传递两个参数，一个是当前活动的对象指针作为上下文的对象，另一个是Java的一个类型，传递这两个参数叫做显示intent。显示就是表明了做事的意图，这句话表明了要启动一个活动，这个活动的类的名字就是MessageActivity。隐式intent没有明确告诉要启动哪个活动，隐式中传递的就不是Java的一个类，而是一个动作action。StartActivity用于启动一个服务，如果要使用具有返回值的启动，则可以使用StartActivityForResult()方法来实现。   1. MessageActivity.java   String message = intent.getStringExtra(MainActivity.MESSAGE\_STRING);  从MainActivity.java中的putExtra的方法中使用getStringExtra中取得一个字符串的属性。  实验结果如下：    图1. 软件启动消息发送的主页面    图2. 显示消息详细弹窗的页面    图3. 跳转活动后的页面 |
| **三、实验总结**  在本次实验中，学会了关于.xml的文件结构布局，学会如何去启动一个新的活动。实验过程中，也遇到了一些问题，一开始是以为在布局的实现当中可以按照指导书上的代码进行布局，没有考虑到顺序的影响同样会影响布局，就把顺序打乱了发现显示出来的效果和指导书上的效果不一致，通过修改为正确的顺序才能正确运行出想要的页面。  在布局过程中，一开始我的想法只是考虑到了整个控件上下左右的边距，距离的数值如何去设定，在运行之后，发现整个页面都很奇怪，比如说输入的文字没有整体在同一条水平线上，因为按钮本来的宽度与输入框的宽度也是不定的，所以在没有绑定约束布局的情况下，两个控件都会集中在页面视图的左上角，通过观察指导书上的代码，我才明白了约束布局的作用及其重要性。  在设置事件侦听器中，主要遇到的问题是不懂具体方法的实现，没有将输入的内容和弹出窗口显示的内容绑定在一起，通过最后在输入框中设置id，再在侦听器中通过绑定id而绑定输入框的内容，最后能够实现。在新增活动当中，是要通过另一个Java类去启动活动，是要通过显式intent来实现调用，通过putExtra和GetExtraString两个方法来获得我们在输入框中输入的内容，通过新建一个布局文件.xml来进行新页面的跳转展示。 |
| **四、实验代码**  1、activity\_message.xml  <?xml version="1.0" encoding="utf-8"?> <androidx.constraintlayout.widget.ConstraintLayout  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".MessageActivity">   <TextView  android:id="@+id/message"  android:hint="Message from MainActivity"  android:textColor="@color/colorAccent"  android:textSize="24sp"  android:gravity="center"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"/> </androidx.constraintlayout.widget.ConstraintLayout>  2、activity\_main.xml  <?xml version="1.0" encoding="utf-8"?> <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".MainActivity">   <EditText  android:id="@+id/message"  android:hint="Enter a message"  android:layout\_marginLeft="16dp"  android:layout\_marginTop="16dp"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  app:layout\_constraintEnd\_toEndOf="@+id/send\_message"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent"/>   <Button  android:id="@+id/send\_message"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="16dp"  android:text="Send Message"  app:layout\_constraintBaseline\_toBaselineOf="@+id/message"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toEndOf="@+id/message"  />  </androidx.constraintlayout.widget.ConstraintLayout>  3、MessageActivity.java  package com.example.cord04\_message;  import androidx.appcompat.app.AppCompatActivity;  import android.content.Intent; import android.os.Bundle; import android.widget.TextView;  public class MessageActivity extends AppCompatActivity {   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_message);   TextView tvMessage = findViewById(R.id.message);  Intent intent = getIntent();  String message = intent.getStringExtra(MainActivity.MESSAGE\_STRING);  if(message!=null){  if(tvMessage!=null){  tvMessage.setText(message);  }  }   } }   1. MainActivity.java   package com.example.cord04\_message;  import androidx.appcompat.app.AppCompatActivity;  import android.content.Intent; import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.Toast;  public class MainActivity extends AppCompatActivity {  public static final String MESSAGE\_STRING ="com.example.Cord04\_message";   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);   final EditText etMessage = findViewById(R.id.message);  Button btSend = findViewById(R.id.send\_message);   btSend.setOnClickListener(new View.OnClickListener(){  @Override  public void onClick(View view){  String message = etMessage.getText().toString();  Intent intent = new Intent(MainActivity.this,MessageActivity.class);  intent.putExtra(MESSAGE\_STRING,message);  startActivity(intent);  }  }); // btSend.setOnClickListener(new View.OnClickListener(){ // @Override // public void onClick(View view){ // String message = etMessage.getText().toString(); // Toast.makeText(MainActivity.this,message,Toast.LENGTH\_SHORT).show(); // } // });  } }  **（第五小节）**   |  | | --- | | **一、实验目的**  1、掌握activity的生命周期的概念  2、学会针对activity不同生命周期中对应的回调方法  3、通过回调方法来实现保存数据和恢复数据 | | **二、实验内容**  代码分析：   1. MainActivity.java   protected void onSaveInstanceState(@NonNull Bundle outState) {  outState.putInt(COUNT\_VALUE,count);  super.onSaveInstanceState(outState);  protected void onRestoreInstanceState(@NonNull Bundle savedInstanceState) {  super.onRestoreInstanceState(savedInstanceState);  count = savedInstanceState.getInt(COUNT\_VALUE);  if(tvCount!=null){  tvCount.setText(Integer.toString(count));  }   这两个方法都是接收Bundle类型的参数，onSaveInstanceState方法中，是通过outState来保存数据，在保存的时候super调用父类的方法是写在保存数据之后；在恢复数据onRestoreInstanceState方法中，super调用父类的方法是放在最开头。通过putInt来存储数据，通过getInt来恢复数据。getInt方法中的defaultValue是默认数据，如果没有存储任何数据，就返回一个默认的数据。恢复数据的时候，要将数据显示在显示屏上，要通过setText来显示存储的数据，这样子就可以实现在旋转屏幕的时候，重新创建MainActivity后数据能够得以保存重新显示在屏幕上。  实验结果如下：    图4. 计数器正常显示页面    图5. 屏幕翻转后计数器的页面 | | **三、实验总结**  在本次实验中，主要就是利用实验一设计的计数器，再加一个保存和恢复数据的方法。在每次的activity.xml被调用的时候都是重新开始运行，如果在重新运行之前能够保存有数据，则重新调用不会改变数据的数值，而是根据保存的数值继续运行。因为在实验一中的布局实现整体来说思路比较清晰，所以主要遇到的问题则是在Bundle类型参数的理解上，在第一次输入代码的时候，没有考虑到onRestoreInstanceState方法中的Bundle对象要进行判断是否为空，通过查询相关资料，结合老师课上讲的内容，才懂得因为onSaveInstanceState 不一定会被调用，所以onCreate()里的Bundle参数可能为空，如果使用onCreate()来恢复数据，一定要做非空判断。而onRestoreInstanceState的Bundle参数一定不会是空值，因为它只有在上次activity被回收了才会调用。 | | **四、实验代码**  1、activity\_main.xml  <?xml version="1.0" encoding="utf-8"?> <LinearLayout  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"  tools:context=".MainActivity">   <Button  android:id="@+id/btnShowToast"  android:text="Show Toast"  android:textColor="@android:color/white"  android:background="@color/colorPrimary"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"/>   <TextView  android:id="@+id/tvCount"  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:text="0"  android:layout\_weight="1"  android:gravity="center"  android:textSize="160dp"  android:textColor="@color/colorAccent" />   <Button  android:id="@+id/btnCount"  android:text="Count"  android:textColor="@android:color/white"  android:background="@color/colorPrimary"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"/> </LinearLayout> |   2、MainActivity.java  package com.example.cord02\_jishu;  import androidx.annotation.NonNull; import androidx.appcompat.app.AppCompatActivity;  import android.os.Bundle; import android.text.InputType; import android.view.View; import android.widget.Button; import android.widget.TextView; import android.widget.Toast;  import java.util.Locale;  public class MainActivity extends AppCompatActivity {  private static final String COUNT\_VALUE = "count\_value";  private int count = 0;  TextView tvCount;   @Override  protected void onSaveInstanceState(@NonNull Bundle outState) {  outState.putInt(COUNT\_VALUE,count);  super.onSaveInstanceState(outState);   }  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  Button showToast = findViewById(R.id.btnShowToast);  Button countButton = findViewById(R.id.btnCount);  tvCount = findViewById(R.id.tvCount);  showToast.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  Toast.makeText(MainActivity.this,"Hello World!",Toast.LENGTH\_SHORT).show();  }  });  countButton.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  tvCount.setText(Integer.toString(++count));  }  });   }   @Override  protected void onRestoreInstanceState(@NonNull Bundle savedInstanceState) {  super.onRestoreInstanceState(savedInstanceState);  count = savedInstanceState.getInt(COUNT\_VALUE);  if(tvCount!=null){  tvCount.setText(Integer.toString(count));  }  } }  **（第六小节）**   |  | | --- | | **一、实验目的**  1、学会布局的布置、使用方式  2、理解并使用Vector Asset资源  3、掌握Image View、View | | **二、实验内容** | | 代码分析：  1、btn\_bg\_pressed.xml  <?xml version="1.0" encoding="utf-8"?> <shape xmlns:android="http://schemas.android.com/apk/res/android"> <corners android:radius="24dp"/>  <solid android:color="#5E2ECF"/> </shape>  btn\_bg\_normal.xml  <?xml version="1.0" encoding="utf-8"?> <shape xmlns:android="http://schemas.android.com/apk/res/android">  <corners android:radius="24dp"/>  <gradient  android:angle="0"  android:startColor="#FFFF0000"  android:endColor="#80FF00FF"/> </shape>  btn\_bg\_selector  <?xml version="1.0" encoding="utf-8"?> <selector xmlns:android="http://schemas.android.com/apk/res/android"> <item android:drawable="@drawable/btn\_bg\_pressed"  android:state\_focused="true"/>  <item android:drawable="@drawable/btn\_bg\_normal"  android:state\_focused="true"/>  <item android:drawable="@drawable/btn\_bg\_normal"/> </selector>  可以通过这些代码来设置控件的不同样式，通过在activity.xml中引用这些具体的样式形式，就可以得到不一样的结果，所以一般我们在设计的时候，都是在drawable文件夹中新建资源来进行修改样式。  final ImageView ivPwdSwitch = findViewById(R.id.iv\_pwd\_switch);  etPwd = findViewById(R.id.et\_pwd);;   ivPwdSwitch.setOnClickListener(new View.OnClickListener(){  @Override  public void onClick(View view){  bPwdSwitch = !bPwdSwitch;  if(bPwdSwitch){  ivPwdSwitch.setImageResource(R.drawable.ic\_visibility\_black\_24dp);  etPwd.setInputType(InputType.TYPE\_TEXT\_VARIATION\_VISIBLE\_PASSWORD);  }else {  ivPwdSwitch.setImageResource(R.drawable.ic\_visibility\_off\_black\_24dp);  etPwd.setInputType(InputType.TYPE\_TEXT\_VARIATION\_VISIBLE\_PASSWORD | InputType.TYPE\_CLASS\_TEXT);  etPwd.setTypeface(Typeface.DEFAULT);  }  } }  以上部分代码用于实现点击所导入的图片，进行密码的明文和暗文的变化。一开始我们导入的图片是属于暗文显示，所以在代码的if中首先定义的就是字符类型为password类型，接着如果我们点击这个图标，就会将这个图标更换为明文显示的图标，并且将字符类型变为正常输入显示的类型。  实验运行结果截图：    图6. 登录页面（密码为密文）    图7. 登录页面（密码为明文）  **三、实验总结**  在本次实验中，主要是学会了如何去组织这个页面的布局，并且导入Android studio中自带的图片资源库，如何设置在应用列表当中的启动图标。实验操作过程中，主要遇到的问题在于实现密码明文暗文切换上，一开始是不知道如何使用点击一个图标进行绑定事件，学过前端，对于不是按钮的事件绑定还是有点模糊，直到联想到前面实验的内容，想到可以利用id来绑定事件，从而能够成功绑定。在点击之后更换明文暗文图标的时候，同样也遇到了问题，不知道如何去将图标也进行改变，只会改变字符的显示类型，通过查询相关的资料还有通过查看指导书，发现了直接setImageResource可以实现更换图片的功能，相当于点击之后发生的事件，将一个新的图片放上去之后再改变字符显示的类型，最终能够顺利实现。 | | **四、实验代码**  1、activity.xml  <?xml version="1.0" encoding="utf-8"?> <LinearLayout  xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".MainActivity"  android:orientation="vertical">   <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:layout\_marginTop="48dp">  <ImageView  android:layout\_width="100dp"  android:layout\_height="100dp"  android:src="@mipmap/logo"  android:layout\_centerInParent="true"/>  </RelativeLayout>   <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:layout\_marginTop="48dp">  <ImageView  android:layout\_width="32dp"  android:layout\_height="32dp"  android:layout\_marginBottom="4dp"  android:layout\_alignParentBottom="true"  android:src="@drawable/ic\_person\_outline\_black\_24dp"/>   <EditText  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="0dp"  android:layout\_alignParentBottom="true"  android:layout\_gravity="center"  android:inputType="text"  android:hint="Email/Account"  android:layout\_marginLeft="48dp"  android:layout\_marginRight="48dp"  android:maxLines="1"/>  </RelativeLayout>   <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="40dp"  android:layout\_marginTop="16dp">  <ImageView  android:layout\_width="32dp"  android:layout\_height="32dp"  android:layout\_marginBottom="0dp"  android:layout\_alignParentBottom="true"  android:src="@drawable/ic\_lock\_outline\_black\_24dp"/>  <EditText  android:id="@+id/et\_pwd"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="0dp"  android:layout\_alignParentBottom="true"  android:inputType="textPassword"  android:hint="Password"  android:layout\_gravity="center"  android:layout\_marginLeft="48dp"  android:layout\_marginRight="48dp"  android:maxLines="1" />  <ImageView  android:id="@+id/iv\_pwd\_switch"  android:layout\_width="24dp"  android:layout\_height="24dp"  android:layout\_marginBottom="6dp"  android:layout\_marginRight="8dp"  android:layout\_alignParentBottom="true"  android:layout\_alignParentRight="true"  android:src="@drawable/ic\_visibility\_off\_black\_24dp"/>  </RelativeLayout>   <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_weight="1"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="32dp">  <Button  android:text="Login"  android:textColor="@android:color/white"  android:layout\_marginTop="32dp"  android:layout\_alignParentTop="true"  android:layout\_marginLeft="32dp"  android:layout\_marginRight="32dp"  android:background="@drawable/btn\_bg\_selector"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"/>   </RelativeLayout>  </LinearLayout>   1. MainActivity.java   package com.example.cord05\_login;  import androidx.appcompat.app.AppCompatActivity;  import android.graphics.Typeface; import android.os.Bundle; import android.text.InputType; import android.view.View; import android.widget.EditText; import android.widget.ImageView;  public class MainActivity extends AppCompatActivity {   private Boolean bPwdSwitch = false;  private EditText etPwd;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  final ImageView ivPwdSwitch = findViewById(R.id.iv\_pwd\_switch);  etPwd = findViewById(R.id.et\_pwd);;   ivPwdSwitch.setOnClickListener(new View.OnClickListener(){  @Override  public void onClick(View view){  bPwdSwitch = !bPwdSwitch;  if(bPwdSwitch){  ivPwdSwitch.setImageResource(R.drawable.ic\_visibility\_black\_24dp);  etPwd.setInputType(InputType.TYPE\_TEXT\_VARIATION\_VISIBLE\_PASSWORD);  }else {  ivPwdSwitch.setImageResource(R.drawable.ic\_visibility\_off\_black\_24dp);  etPwd.setInputType(InputType.TYPE\_TEXT\_VARIATION\_VISIBLE\_PASSWORD | InputType.TYPE\_CLASS\_TEXT);  etPwd.setTypeface(Typeface.DEFAULT);  }  }  });  }  }  3、btn\_bg\_normal.xml  <?xml version="1.0" encoding="utf-8"?> <shape xmlns:android="http://schemas.android.com/apk/res/android">  <corners android:radius="24dp"/>  <gradient  android:angle="0"  android:startColor="#FFFF0000"  android:endColor="#80FF00FF"/> </shape>  4、btn\_bg\_pressed.xml  <?xml version="1.0" encoding="utf-8"?> <shape xmlns:android="http://schemas.android.com/apk/res/android"> <corners android:radius="24dp"/>  <solid android:color="#5E2ECF"/> </shape>  5、btn\_bg\_selector  <?xml version="1.0" encoding="utf-8"?> <selector xmlns:android="http://schemas.android.com/apk/res/android"> <item android:drawable="@drawable/btn\_bg\_pressed"  android:state\_focused="true"/>  <item android:drawable="@drawable/btn\_bg\_normal"  android:state\_focused="true"/>  <item android:drawable="@drawable/btn\_bg\_normal"/> </selector>  **（第七小节）**   |  | | --- | | **一、实验目的**  1、能够了解并使用ListView控件  2、掌握ListView控件中自定义Item布局的使用方法  3、了解并使用ArryAdapter，掌握其自定义方法  4、掌握CardView控件的基本使用方法 | | **二、实验内容** | | 代码分析：   1. MainActivity.java   titles = getResources().getStringArray(R.array.titles);  authors = getResources().getStringArray(R.array.authors);  images = getResources().obtainTypedArray(R.array.images);  用于获取在arrays.xml中定义的资源，类型为字符串数组  NewsAdapter newsAdapter =  new NewsAdapter(MainActivity.this,R.layout.list\_item,newsList);  ListView listView = findViewById(R.id.lv\_news\_list);  listView.setAdapter(newsAdapter);  adapter属于桥梁，响应用户的交互。在用户进行在页面中上下滑动的时候，就会调用adapter中的getview方法去获取新的资源来进行显示。参数第二个是自定义的组件。  2、NewsAdapter.java  public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent)  view = LayoutInflater.from(getContext()).inflate(resource,parent,false);  参数position是listview传递的，代表对应的一个新闻对象。LayoutInflater用于布局的，会把布局文件实例化，通过ViewGroup传递的参数取布局的大小。  实验运行结果截图：    图8. 新闻页面1    图9. 新闻页面2  **三、实验总结**  在本次实验中，主要学会了如何使用cardview布局来进行设置里面的具体内容，还有将常见的新闻列表中的标题、作者和图片等相关的资源在代码中具体调用实现在界面上。实验中主要遇到的问题是对于adapter使用的不理解，对于其中对应的参数的具体表达含义不熟悉，导致在写代码的过程中要依赖于各种参考的文章还有指导书，不能很好地理解其具体表达的含义，还有就是使用cardview布局中，修改build.gradle的时候，根据指导书上的内容不能够成功配置，通过找到网上的一些相关方法，将Android studio的一些参数版本进行相对应升级之后，在代码中也能对应上升级后的配置参数，下载对应版本的虚拟机，最终才能够实现cardview布局。 | | **四、实验代码**  1、MainActivity.java  package com.example.newslist;  import androidx.appcompat.app.AppCompatActivity;  import android.content.res.TypedArray;  import android.os.Bundle;  import android.widget.ArrayAdapter;  import android.widget.ListView;  import android.widget.SimpleAdapter;  import java.util.ArrayList;  import java.util.HashMap;  import java.util.List;  import java.util.Map;  public class MainActivity extends AppCompatActivity {  private static final String NEWS\_TITLE = "news\_title";  private static final String NEWS\_AUTHOR = "news\_author";  private List<Map<String,String>> dataList = new ArrayList<>();  private String[] titles = null;  private String[] authors = null;  private List<News> newsList = new ArrayList<>();  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  initData();  NewsAdapter newsAdapter =  new NewsAdapter(MainActivity.this,R.layout.list\_item,newsList);  ListView listView = findViewById(R.id.lv\_news\_list);  listView.setAdapter(newsAdapter);  }  private void initData(){  int length;  titles = getResources().getStringArray(R.array.titles);  authors = getResources().getStringArray(R.array.authors);  images = getResources().obtainTypedArray(R.array.images);  System.out.println(titles.length+"length");  TypedArray images = getResources().obtainTypedArray(R.array.images);  if(titles.length>authors.length){  length=authors.length;  }  else{  length = titles.length;  }  for (int i=0;i<length;i++){  News news = new News();  news.setTitle(titles[i]);  news.setAuthor(authors[i]);  news.setImageId(images.getResourceId(i,0));  newsList.add(news);  }  }  } |  1. **News.java**   package com.**example**.newslist;  public class News {  private String title;  private String author;  private String content;  private int imageId;  public String getTitle() {  return title;  public void setTitle(String title) {  this.title = title;  }  public String getAuthor() {  return author;  public void setAuthor(String author) {  this.author = author;  public String getContent() {  return content;  }  public void setContent(String content) {  this.content = content;  }  public int getImageId() {  return imageId;  }  public void setImageId(int imageId) {  this.imageId = imageId;  }  }   1. NewsAdapter.java   package com.example.newslist;  import android.content.Context;  import android.view.LayoutInflater;  import android.view.View;  import android.view.ViewGroup;  import android.widget.ArrayAdapter;  import android.widget.ImageView;  import android.widget.TextView;  import androidx.annotation.NonNull;  import androidx.annotation.Nullable;  import java.util.List;  public class NewsAdapter extends ArrayAdapter<News> {  private List<News> objects;  private Context context;  private int resource  public NewsAdapter(@NonNull Context context, int resource, @NonNull List<News> objects) {  super(context, resource, objects);  this.context = context;  this.objects = objects;  this.resource = resource;  }  @NonNull  @Override  public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {  News news = getItem(position);  View view;  view = LayoutInflater.from(getContext()).inflate(resource,parent,false);  TextView title = view.findViewById(R.id.title);  TextView author = view.findViewById(R.id.author);  ImageView imageView = view.findViewById(R.id.image);  title.setText(news.getTitle());  author.setText(news.getAuthor());  imageView.setImageResource(news.getImageId());  return view;  }  } |  1. activity.xml   <?xml version="1.0" encoding="utf-8"?>  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".MainActivity">  <ListView  android:id="@+id/lv\_news\_list"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintLeft\_toLeftOf="parent"  app:layout\_constraintRight\_toRightOf="parent"  app:layout\_constraintTop\_toTopOf="parent" />  </androidx.constraintlayout.widget.ConstraintLayout>   1. list\_item.xml   <?xml version="1.0" encoding="utf-8"?>  <androidx.cardview.widget.CardView android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  xmlns:app="http://schemas.android.com/apk/res-auto"  android:padding="20dp"  app:cardCornerRadius="20dp"  xmlns:android="http://schemas.android.com/apk/res/android">  <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content">  <ImageView  android:layout\_width="match\_parent"  android:layout\_height="192dp"  android:scaleType="centerCrop"  android:id="@+id/image"/>  <TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  style="@style/TextAppearance.AppCompat.Subhead"  android:id="@+id/title"  android:layout\_below="@+id/image"  android:paddingTop="8dp"  android:paddingStart="8dp"  android:paddingBottom="4dp"  android:textColor="@color/black"  android:textSize="18sp"  android:textStyle="bold"/>  <TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/author"  android:layout\_below="@+id/title"  android:paddingStart="8dp"  android:paddingBottom="4dp"  android:textColor="@color/purple\_500"  style="@style/TextAppearance.AppCompat.Subhead"  android:textSize="14sp"  />  </RelativeLayout>  </androidx.cardview.widget.CardView> |