

第 5 章

列表控件

在现实生活中,人们经常会使用 QQ、微信等应用程序。在这些应用程序中通常会有一个页面展示多个条目,并且每个条目的布局都是一样的。如果利用前面所学习的知识实现这种布局,需要创建大量相同的布局,使用这种方式并不利于程序维护和扩展。针对上述情况,Android 系统提供了功能强大的列表控件,通过列表控件可轻松实现上述需求。

列表控件是 Android 系统开发中使用最广泛的控件之一,常见的列表控件包括 Spinner(下拉列表)、ListView(普通列表)、GridView(网格列表)、RecyclerView(增强列表)等。通过列表控件可展示多项数据,并且开发者可动态配置数据源,列表控件可根据所适配的数据源不同展示不同的内容。Android 中主要采用适配器模式帮助建立列表控件和数据源之间的联系。因此在使用列表控件的时候,还需要创建适配器对象并为适配器提供数据源。Adapter(适配器)对象用来指明数据源中的每一项数据在列表控件中如何显示。列表控件调用 setAdapter()方法把 Adapter 对象传递进来,即可将数据显示在列表中。

本章通过对“欢乐购商城”分析,帮助读者学会使用列表控件。通过对不同种类的列表控件综合运用,实现“欢乐购商城”中首页、商品列表、订单列表等页面的功能。



学习目标

本章要点

- (1) 掌握下拉列表 Spinner 的功能和用法。
- (2) 掌握普通列表 ListView 的功能和用法。
- (3) 掌握网格列表 GridView 的功能和用法。
- (4) 掌握增强列表 RecyclerView 的功能和用法。



视频详解



5.1 下拉列表 Spinner



任务陈述

在“欢乐购商城”项目中修改收货地址页面使用到了 Spinner 控件,下面对修改收货地址页面做概要分析。

分析：在修改用户收货地址页中，需要用户填写地址。在填写地址时需要选择不同的省份市区。由于省份等信息是固定不变的，用户只需从一组数据中选择，因此在这里采用 Spinner 控件实现，当用户单击此控件时，下拉列表中显示出所有省份信息选项供用户选择，如图 5-1 所示。



图 5-1 修改地址页面



相关知识

5.1.1 Spinner 控件

下拉列表 Spinner 类似于下拉菜单，默认情况下展示列表项中的第一项，单击 Spinner 控件时会弹出一个包含所有数据的下拉列表。Spinner 比较节省空间，常用于固定值选择或者条件筛选等。例如，在用户输入地址信息的时候，在选择省份或者地区时通常可以采用 Spinner 列表控件提供给用户，让用户从中选择，这样可减少用户的输入和避免用户输入错误信息。

使用列表控件的关键步骤如下。

- (1) 在布局文件中添加列表控件，在 Java 代码文件中通过 id 属性获取到列表控件。
- (2) 准备数据源，数据源可以是数组或者集合。
- (3) 构建 Adapter 对象，指定列表中每一项数据的显示样式，并将数据源数据通过构造函数等方式传递给 Adapter 对象。
- (4) 列表控件调用 setAdapter() 方法，关联创建好的 Adapter 对象，展示数据源中的数据。

下面就以选择编程语言为例，讲解 Spinner 控件的具体用法。

- (1) 创建名字为 chart0501 的应用程序，指定包名为 com. example. administrator.



视频详解

chart0501。

(2) 设置布局文件,在 res/layout/activity_main.xml 文件中添加一个 TextView 用于显示标题,再添加一个 Spinner 控件用于显示下拉列表。布局文件的核心代码如下。

程序清单 5-1: chart0501\app\src\main\res\layout\activity_main.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="wrap_content"
6      android:orientation="horizontal"
7      android:gravity="center_vertical"
8      tools:context=".MainActivity">
9      <TextView
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:text="你学习的编程语言:"
13         android:textSize="16sp"/>
14      <Spinner
15         android:id="@+id/spinner"
16         android:layout_width="wrap_content"
17         android:layout_height="wrap_content"/>
18  </LinearLayout>

```

→ Spinner 添加 id 属性
→ 为 Spinner 设置宽度
→ 为 Spinner 设置高度

(3) Spinner 列表由若干 Item 组成,每个 Item 显示编程语言的名称,因此需要设置 Item 选项的布局。在 res/layout 文件夹中创建一个 Item 界面的布局文件 simple_list_item.xml,在该文件中添加一个 TextView 用来展示每项的内容。注意:不需要在 TextView 的外层添加布局,完整的布局文件代码如下。

程序清单 5-2: chart0501\app\src\main\res\layout\simple_list_item.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <TextView xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="wrap_content"
4      android:layout_height="wrap_content"
5      android:padding="5dp"
6      android:textSize="16sp"/>

```

(4) 编写界面交互代码,在 MainActivity 中定义一个数组 programLang,存储 Spinner 下拉列表中显示的编程名称,并创建 ArrayAdapter 对象,调用 Spinner 控件的 setAdapter() 方法,将 Adapter 与列表控件关联起来,实现数据适配,具体代码如下。

程序清单 5-3: chart0501\java\com\example\administrator\chart0401\MainActivity.java

```

1  public class MainActivity extends AppCompatActivity {
2      private String[] programLang = {"Java", "C#",
3          "Python", "Javascript", "C++", "C"};
4      private Spinner spinner;

```

→ 定义数组保存编程语言

```
5         @Override
6         protected void onCreate(Bundle savedInstanceState) {
7             super.onCreate(savedInstanceState);
8             setContentView(R.layout.activity_main);
9             spinner = (Spinner) findViewById(R.id.spinner);    →获取 Spinner 对象
10            ArrayAdapter<String> stringArrayAdapter = new ArrayAdapter<String>
11                (this, R.layout.simple_list_item, programLang);    →创建适配器对象
12            spinner.setAdapter(stringArrayAdapter);    →为 Spinner 对象绑定适配器对象
13
14        }
15    }
```

上述代码中,第2~3行代码定义一个数组 `programLang`,存储 `Spinner` 下拉列表中显示的编程名称。第10行代码定义一个 `ArrayAdapter` 对象,并传入三个参数,第一个参数表示当前对象,第二个参数表示每个 `Item` 的布局文件,第三个参数表示数据源。第13行代码通过 `setAdapter()` 方法为 `Spinner` 控件设置适配器。

上述代码的运行效果如图 5-2 所示。

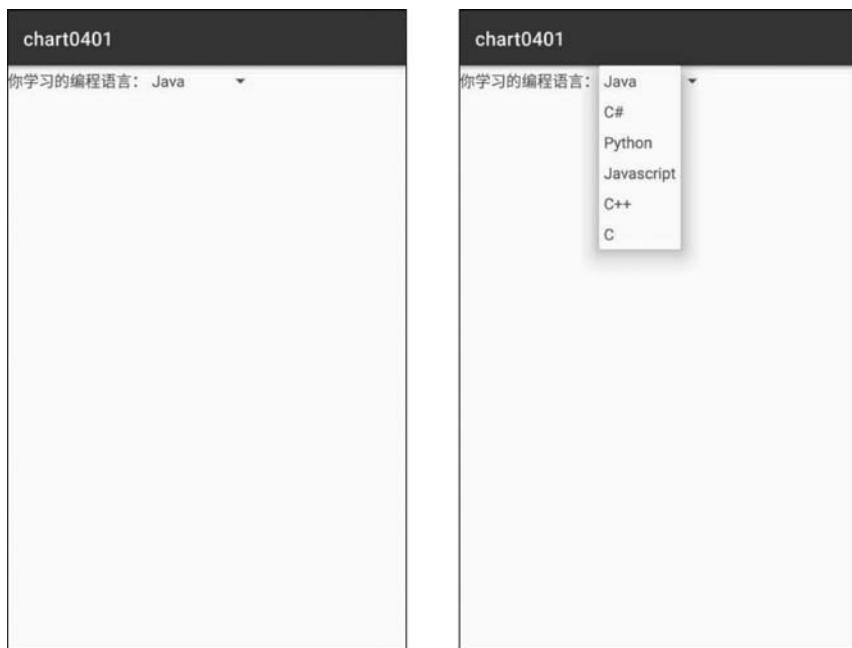


图 5-2 程序运行效果

5.1.2 Adapter 适配器

在为 `Spinner` 控件添加数据的时候用到了数据适配器 (`Adapter`),数据适配器是数据和视图之间的桥梁,类似于一个转换器,将复杂的数据转换成用户设定的样式呈现出来,在 `Android` 系统中提供了多种适配器,它们之间的关系如图 5-3 所示。



视频详解

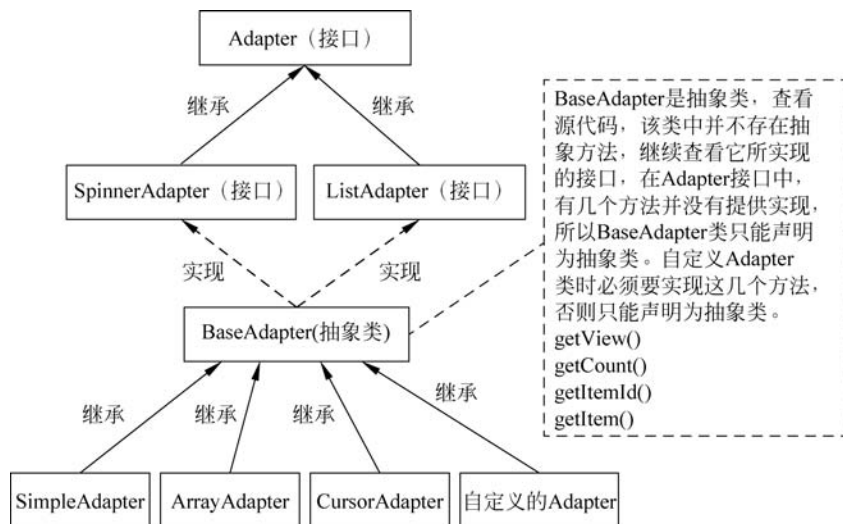


图 5-3 Adapter 相关的接口

从图 5-3 可知，实现 Adapter 相关接口的基类是 BaseAdapter，它实现了接口里的大部分方法，对于少数需要根据具体情景才能确定的方法则没有实现，因此它被声明为不能实例化。针对一些常用情景系统提供了三个子类：ArrayAdapter、SimpleAdapter、CursorAdapter。在使用时需要自定义一个 Adapter 继承自 BaseAdapter，并重写里面的 getView()、getCount()、getItem()、getItemId() 方法。自定义 Adapter 的优点是可以使数据按照设定的形式显示，非常灵活，缺点是代码相对较多，需要用户自己重写各个方法。

(1) ArrayAdapter：默认情况下只能显示文本，如果想显示其他的 View 控件，比如 ImageView，需要重写 getView() 方法。通常，ArrayAdapter 的数据源是数组或者集合的形式。

ArrayAdapter 有多个构造函数，以其中两个举例说明：

```
public ArrayAdapter (Context context, int resource, int textViewResourceId, T[] objects);
```

```
public ArrayAdapter (Context context, int resource, List<T> objects)。
```

ArrayAdapter() 构造方法中的 5 个参数含义如下。

- ① Context：context 上下文对象。
- ② resource：Item 布局的资源 id。
- ③ textViewResourceId：Item 布局中相应 TextView 的 id。
- ④ T[] objects：需要适配的数组类型的数据。
- ⑤ List<T> objects：需要适配的 List 类型的数据。

(2) SimpleAdapter：可用于显示列表项相对复杂的列表，要求所有的列表项结构相同，显示样式相同。SimpleAdapter 继承自 BaseAdapter 并实现 BaseAdapter 的四个抽象方法并对其进行封装。因此在使用 SimpleAdapter 进行数据适配时，只需要在构造方法中传入相应的参数即可。

SimpleAdapter 的构造方法的具体信息如下。

```
public SimpleAdapter (Context context, List<? extends Map<String, ? >> data, int resource, string[] from, int[] to)
```

SimpleAdapter()构造方法中的 5 个参数的含义如下。

- ① context: 表示上下文对象。
- ② data: 数据集合, data 中的每一项对应 ListView 控件中的条目的数据。
- ③ resource: Item 布局的资源 id。
- ④ from: Map 集合中的 key 值。
- ⑤ to: Item 布局中对应的控件。

(3) BaseAdapter: 本质上是一个抽象类, 通常在自定义适配器时会继承 BaseAdapter。该类有四个抽象方法, 在使用自定义适配器时需要实现这几个抽象方法, 如表 5-1 所示。

表 5-1 BaseAdapter 类抽象方法

方 法 名	功 能 描 述
public int getCount()	获取 Item 条目的总数
public object getItem(int position)	根据 position(位置)获取某个 Item 对象
public long getItemId(int position)	根据 position(位置)获取某一个 Item 的 id
public View getView (int position, View convertView, ViewGroup parent)	获取相应 position 对应的 Item 视图

5.1.3 实战演练——收货地址

通过对 Spinner 的学习, 我们可以对“欢乐购商城”中修改地址页面进行实现。布局页面的代码如下。



视频详解

程序清单 5-4: chart0502\app\src\main\res\layout\activity_main.xml

```
1  <?xml version = "1.0" encoding = "utf-8"?>
2  <RelativeLayout xmlns:android = "http://schemas.android.com/apk/res/android"
3      xmlns:tools = "http://schemas.android.com/tools"
4      android:layout_width = "match_parent"
5      android:layout_height = "match_parent"
6      tools:context = ".MainActivity">
7      <EditText
8          android:id = "@ + id/edit_name"
9          android:layout_width = "match_parent"
10         android:layout_height = "wrap_content"
11         android:layout_marginTop = "5dp"
12         android:hint = "姓名" />
13     <EditText
14         android:id = "@ + id/edit_phone"
15         android:layout_width = "match_parent"
16         android:layout_height = "wrap_content"
17         android:layout_below = "@ + id/edit_name"
18         android:layout_marginTop = "5dp"
19         android:hint = "电话" />
```

```

20     <TextView
21         android:id="@+id/tv_proName"
22         android:layout_width="wrap_content"
23         android:layout_height="30dp"
24         android:layout_below="@+id/edit_phone"
25         android:layout_marginTop="5dp"
26         android:gravity="center"
27         android:text="省" />
28     <Spinner
29         android:id="@+id/provinceSpinner"      →Spinner 添加 id 属性
30         android:layout_width="wrap_content"    →为 Spinner 设置宽度
31         android:layout_height="30dp"          →为 Spinner 设置高度
32         android:layout_alignTop="@+id/tv_proName"
33         android:layout_marginLeft="10dp"
34         android:layout_toRightOf="@+id/tv_proName">
35     </Spinner>
36     <TextView
37         android:id="@+id/tv_cityName"
38         android:layout_width="wrap_content"
39         android:layout_height="30dp"
40         android:layout_alignBottom="@+id/provinceSpinner"
41         android:layout_marginTop="5dp"
42         android:layout_toRightOf="@+id/provinceSpinner"
43         android:text="市"
44         android:gravity="center"
45         android:textSize="13sp" />
46     <Spinner
47         android:id="@+id/citySpinner"
48         android:layout_width="wrap_content"
49         android:layout_height="30dp"
50         android:layout_alignBottom="@+id/tv_cityName"
51         android:layout_toRightOf="@+id/tv_cityName"/>
52     <TextView
53         android:id="@+id/tv_countryName"
54         android:layout_width="wrap_content"
55         android:layout_height="30dp"
56         android:layout_alignBottom="@+id/citySpinner"
57         android:layout_marginTop="5dp"
58         android:layout_toRightOf="@+id/citySpinner"
59         android:gravity="center"
60         android:text="县" />
61     <Spinner
62         android:id="@+id/countrySpinner"
63         android:layout_width="wrap_content"
64         android:layout_height="30dp"
65         android:layout_alignBottom="@+id/tv_countryName"
66         android:layout_toRightOf="@+id/tv_countryName"/>
67     <EditText
68         android:id="@+id/edit_detailAddress"
69         android:layout_width="match_parent"

```

```
69         android:layout_height = "100dp"
70         android:layout_below = "@ + id/tv_proName"
71         android:layout_marginTop = "10dp"
72         android:background = "@drawable/textview_borders"
73         android:gravity = "top"
74         android:minLines = "5"
75         android:textSize = "16sp"
76         android:text = "中华路 58 号"/>
77     </RelativeLayout>
```

spinner_item.xml 的布局文件如下。

程序清单 5-5: char0502\app\src\main\res\layout\spinner_item.xml

```
1     <?xml version = "1.0" encoding = "utf - 8"?>
2     <TextView xmlns:android = "http://schemas.android.com/apk/res/android"
3         android:layout_width = "match_parent"
4         android:layout_height = "wrap_content"
5         android:padding = "5dp"
6         android:textSize = "14sp" />
```

MainActivity 页面的代码如下。

程序清单 5-6: char0502app\src\main\res\layout\activity_main.xml

```
1     public class MainActivity extends AppCompatActivity {
2         private EditText edit_name;
3         private EditText edit_phone;
4         private Spinner proPinner;
5         private EditText detailAddress;
6         private Spinner citySpinner;
7         private Spinner countrySpinner;
8         private String provinceName;
9         private String cityName;
10        private String[] partCitys;
11        private String[] partCountry;
12        private String[] province = {"江西省", "河南省", "湖北省"};
13        @Override
14        protected void onCreate(Bundle savedInstanceState) {
15            super.onCreate(savedInstanceState);
16            setContentView(R.layout.activity_main);
17            initView();
18            initData();
19        }
20        private void initView() { →创建方法初始化控件
21            edit_name = (EditText) findViewById(R.id.edit_name);
22            edit_phone = (EditText) findViewById(R.id.edit_phone);
23            proPinner = (Spinner) findViewById(R.id.proviceSpinner);
24            citySpinner = (Spinner) findViewById(R.id.citySpinner);
25            countrySpinner = (Spinner) findViewById(R.id.countrySpinner);
```

```
26         detailAddress = (EditText) findViewById(R.id.edit_detailAddress);
27     }
28     private void initData() {
29         edit_name.setText("李玲");    →为 edit_name 设置数据
30         edit_phone.setText("13789325643");    →为 edit_phone 设置数据
31         detailAddress.setText("气象路 58 号");
32         proPinner.setAdapter(new ArrayAdapter<String>(this, R.layout.spinner_
                                     item, province));
                                     →创建适配器对象,并为 proPinner 绑定该对象
33         proPinner.setOnItemClickListener(new AdapterView.OnItemClickListener() {
                                     →为 proPinner 设置项被选择的事件
34             @Override
35             public void onItemClick(AdapterView<?> parent, View
36             view, int position, long id) {
37                 provinceName = province[position];
38                 switch (position) {
                                     →根据选择不同的省份,加载不同的地区
39                     case 0:
40                         partCitys = new String[]{"南昌", "九江", "赣州", "上饶"};
41                         citySpinner.setAdapter(new ArrayAdapter<String>(MainActivity.this, R.
42                             layout.spinner_item, partCitys));
                                     →创建地区的适配器对象,并为 citySpinner 绑定该对象
43                     break;
44                     case 1:
45                         partCitys = new String[]{"郑州", "南阳", "洛阳", "安阳"};
46                         citySpinner.setAdapter(new ArrayAdapter<String>
47                             (MainActivity.this,
48                             R.layout.spinner_item, partCitys));
49                     break;
50                     case 2:
51                         partCitys = new String[]{"武汉", "黄石", "阳新", "鄂州"};
52                         citySpinner.setAdapter(new ArrayAdapter<String>
53                             ( MainActivity.this, R.layout.spinner_item,
54                             partCitys));
55                     break;
56                 }
57             }
58             @Override
59             public void onNothingSelected(AdapterView<?> parent) {
60             }
61         });
62     }
```

```
63         citySpinner. setOnItemSelectedListener ( new AdapterView.  
                OnItemSelectedListener() {  
                    →为 citySpinner 设置项被选择事件  
64             @Override  
65             public void onItemSelected(AdapterView<?> parent, View  
66                 view, int position, long id) {  
67                 cityName = partCitys[position];  
68                 switch (position) {  
                    →根据选择不同的地区,加载不同的县  
69                     case 0:  
70                         partCountry = new String[] {"红谷滩新区", "东湖区", "西  
                                湖区"};  
71                         countrySpinner.setAdapter(new  
                    →创建县的适配器对象,并为 countrySpinner 绑定该对象  
72                             ArrayAdapter<String>(MainActivity.this, R.layout.  
                                spinner_item,  
73                                 partCountry));  
74                         break;  
75                     case 1:  
76                         partCountry = new String[] {"修水", "永修", "都昌", "德  
安"};  
77                         countrySpinner.setAdapter ( new ArrayAdapter<String>  
                                (MainActivity.this,  
                                    R.layout.spinner_item, partCountry));  
78                         break;  
79  
80                     case 2:  
81                         partCountry = new String[] {"信丰", "大余", "兴国", "于  
都"};  
82                         countrySpinner.setAdapter(new ArrayAdapter<String>(  
83                             MainActivity.this, R.layout.spinner_item, partCountry));  
84                         break;  
85                     }  
86                 }  
87             @Override  
88             public void onNothingSelected(AdapterView<?> parent) {  
89                 }  
90             });  
91         }  
92     }
```

上述程序运行效果如图 5-4 所示。

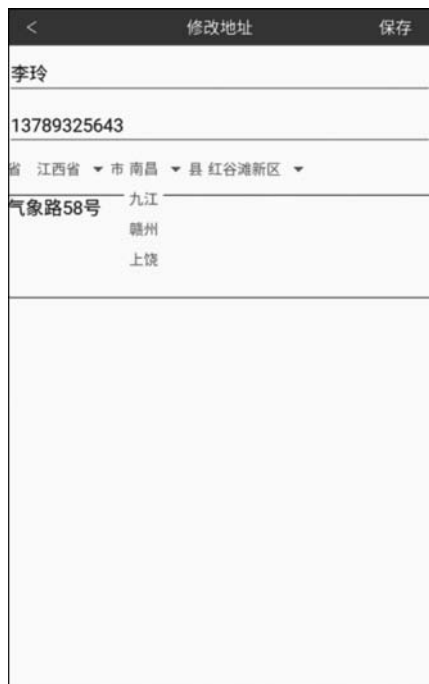


图 5-4 程序运行效果



视频详解



5.2 普通列表 ListView



任务陈述

“欢乐购商城”项目中订单页面使用到了 ListView 控件,下面对修改订单页面做概要分析。

分析: 在订单页面需要展示用户的订单信息,用户的订单信息展示的内容的格式是一致的,有订单商品标题、订单状态、总价和创建时间。对于列表展示并且每一项的格式一致适合采用 ListView 控件实现,如图 5-5 所示。



相关知识



视频详解

5.2.1 ListView 控件

ListView 是使用非常广泛的一种列表控件,它以垂直列表的形式展示所有的数据项,并且能够根据列表的高度自适应屏幕显示,也是早期 Android 开发中用于列表界面开发最多的控件。早期的开发者为了拓展 ListView,对其进行继承和改写,也为其丰富了许多独特的功能,例如,经典的上拉加载新的列表数据,下拉刷新原有的列表数据等。

下面就以展示植物信息为例,讲解 ListView 控件的具体用法。

(1) 创建程序,创建名字为 chart0502 的应用程序。

(2) 导入图片,在 Android Studio 中切换到 Project 选项卡,选中程序中的 res 文件夹,



图 5-5 订单中心

右击选择 New→Directory 选项, 创建一个名为 drawable-hdpi 的文件夹。将程序所需要的图片导入到文件夹。

(3) 设置布局文件, 在 res/layout/activity_main.xml 文件中添加一个 ListView 控件, 并对其相应属性进行设置, 布局文件的核心代码如下。

程序清单 5-7: char0503\app\src\main\res\layout\activity_main.xml

```
1 <ListView
2     android:id="@+id/listView"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:listSelector="#A9A9A9"></ListView>
```

(4) 设置每个 Item 的布局, 在 res/layout 文件夹中创建一个 Item 界面的布局文件 list_item.xml, 在该文件中添加一个 ImageView 用来显示植物的图片, 添加两个 TextView 分别用于展示植物的名称和植物的描述。完整的布局文件代码如下。

程序清单 5-8: chart0503\app\src\main\res\layout\list_item.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5     <ImageView
6         android:id="@+id/iv"
7         android:layout_width="80dp"
8         android:layout_height="80dp"
9         android:layout_margin="5dp"
```

```

10         android:background="@drawable/b"
11         android:scaleType="centerCrop" />
12     <TextView
13         android:id="@+id/tv_title"
14         android:layout_width="match_parent"
15         android:layout_height="wrap_content"
16         android:layout_alignTop="@id/iv"
17         android:layout_toRightOf="@id/iv"
18         android:text="植物名称"
19         android:textSize="18sp"
20         android:textStyle="bold" />
21     <TextView
22         android:id="@+id/tv_depict"
23         android:layout_width="match_parent"
24         android:layout_height="wrap_content"
25         android:layout_below="@id/tv_title"
26         android:layout_marginTop="10dp"
27         android:layout_toRightOf="@id/iv"
28         android:text="植物的描述"
29         android:textSize="16sp" />
30 </RelativeLayout>

```

(5) 在 MainActivity 中创建一个内部类 MyBaseAdapter 继承自 BaseAdapter, 并且在 MyBaseAdapter 中实现对 ListView 控件的适配。完整的代码如下。

程序清单 5-9: chart0503\java\com\example\administrator\chart0503\MainActivity.java

```

1  public class MainActivity extends AppCompatActivity {
2      private ListView mListView;
3      private int[] icons = {R.drawable.a, R.drawable.b,
4          R.drawable.c, R.drawable.d, R.drawable.e, R.drawable.f};
5      private String[] titles = {"白鹃梅", "五色椒", "小檗",
6          "李叶绣线菊", "多花兰", "獐耳细辛"};
7      private String[] depicts =
8          {"白鹃梅又名茧子花、金瓜果等,是蔷薇科白鹃梅属灌木.",
9          "五色椒又名朝天椒,五彩辣椒,为辣椒 变种,味涩",
10         "小檗春日黄花簇簇,秋日红果满枝", "蔷薇科、绣线菊属灌木
11         ,高达3米.", "多花兰是兰科兰属的一种附生植物",
12         "植物獐耳细辛,分布于我国辽宁、安徽、浙江、河南"};
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_main);
17         mListView = (ListView) findViewById(R.id.listView);
18         MyBaseAdapter myBaseAdapter = new MyBaseAdapter();
19         mListView.setAdapter(myBaseAdapter);
20     }
21 }

```

```
22     private class MyBaseAdapter extends BaseAdapter {
23         @Override
24         public int getCount() {
25             return titles.length;
26         }
27         @Override
28         public Object getItem(int position) {
29             return titles[position];
30         }
31         @Override
32         public long getItemId(int position) {
33             return position;
34         }
35         @Override
36         public View getView(int position, View convertView, ViewGroup parent) {
37             View view = View.inflate(MainActivity.this, R.layout.list_item, null);
38             TextView title = (TextView) view.findViewById(R.id.tv_title);
39             TextView depict = (TextView) view.findViewById(R.id.tv_depict);
40             ImageView image = (ImageView) view.findViewById(R.id.iv);
41             image.setBackgroundResource(icons[position]);
42             title.setText(titles[position]);
43             depict.setText(depicts[position]);
44             return view;
45         }
46     }
47 }
```

上述代码中,在第3~12行代码定义了三个数组: titles、depicts 和 icons,分别存储植物的名称、植物的描述和植物的图片,并且这三个数组的长度是一致的。

第22~47行创建了一个 MyBaseAdapter 继承自 BaseAdapter 类,并重写了 BaseAdapter 类中的 getCount()、getItem()、getItemId() 和 getView() 方法。在 getView() 方法中通过 inflate() 方法将 layout.list_item 布局转换成视图对象。并通过 findViewById() 获取到 layout.list_item 布局中的各个控件,最后通过 setText() 和 setBackgroundResource() 方式将文本信息和图片信息展示出来。

该程序的运行效果如图 5-6 所示。

5.2.2 提升 ListView 运行效率

在运行上述代码的时候,当 ListView 上加载的 Item 过多并且快速滑动 ListView 控件的时候,界面会出现卡顿。出现此状况的原因如下。

(1) 当不断滑动 ListView 控件时,就会不断创建 Item 对象。ListView 控件在屏幕上



图 5-6 程序运行效果



视频详解

显示多少个 Item, 就会在适配器 MyBaseAdapter 中的 getView() 中创建多少个 Item 对象。当滑动 ListView 控件时, 滑出屏幕的 Item 对象会被销毁, 新加载到屏幕上的 Item 会创建新的对象, 因此在滑动的过程中就是在不断地创建和销毁 Item 对象。

(2) 在 getView() 中不断执行 findViewById() 方法初始化控件。每当创建一个 Item 对象就会加载一个 Item 布局, 在加载布局的过程中会不断调用 findViewById() 方法初始化控件。这些操作要消耗设备的内存, 因此不断滑动 ListView 就是在不断地初始化控件, 当占用内存过多的时候, 就会出现程序内存溢出异常。

针对上述问题, 需要对 ListView 控件进行优化, 优化的目标是在 ListView 不断滑动过程中不再重复创建 Item 对象, 减少内存的消耗。具体操作如下。

(1) 在 MainActivity 类中创建 ViewHolder 类。

程序清单 5-10: chart0503\java\com\example\administrator\chart0503\MainActivity.java

```
1 public class ViewHolder{
2     TextView tv_title;
3     TextView tv_depict;
4     ImageView image;
5 }
```

(2) 在 MyBaseAdapter 的 getView() 方法中第二个参数 convertView 代表滑出屏幕的 Item 的缓存对象, 当第一次加载 ListView 的时候创建 Item 对象, 当滑动 ListView 控件的时候, 在加载新的 Item 对象的时候可以复用缓存的 convertView 对象。在 getView() 中进行优化, 具体代码如下。

程序清单 5-11: chart0503\java\com\example\administrator\chart0503\MainActivity.java

```
1 public View getView(int position, View convertView, ViewGroup parent) {
2
3
4     ViewHolder viewHolder;
5     if (convertView == null) {
6         convertView = View.inflate(MainActivity.this, R.layout.list_item, null);
7         viewHolder = new ViewHolder();
8         viewHolder.tv_title = (TextView) convertView.findViewById(R.id.tv_title);
9         viewHolder.tv_depict = (TextView) convertView.findViewById(R.id.tv_depict);
10        viewHolder.image = (ImageView) convertView.findViewById(R.id.iv);
11        convertView.setTag(viewHolder);
12    } else {
13        viewHolder = (ViewHolder) convertView.getTag();
14    }
15    viewHolder.image.setBackgroundResource(R.drawable.icons[position]);
16    viewHolder.tv_title.setText(R.string.titles[position]);
17    viewHolder.tv_depict.setText(R.string.depicts[position]);
18    return convertView;
19 }
20 }
```

在上述代码中第 5~18 行中首先判断 convertView 对象是否为 null,如果是 null 表示是第一次加载 Item 项,需要使用 inflate()创建 Item 对象并通过 findViewById()方法找到控件。创建 viewHolder 对象,并将 Item 中的界面控件对象赋值给 viewHolder 对象的属性,最后通过 setTag()和 getTag()方法获取缓存在 convertView 对象中的 ViewHolder 对象。

5.2.3 ListView 的单击事件

在使用 ListView 时,当单击某一个 Item 的时候,可能还需要进行其他操作。例如,在展示商品信息的时候,ListView 可展示商品概要信息,当单击某一商品时,需要跳转到另外一个页面展示商品的详细信息,此时就需要触发 ListView 中 Item 的单击事件。在 ListView 中常用的单击事件分为普通的单击事件和长单击事件,下面以代码的形式说明如何使用单击事件。

在 onCreate()方法中添加以下代码。

程序清单 5-12: chart0503\java\com\example\administrator\chart0503\MainActivity.java

```
1    mListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
2        @Override
3        public void onItemClick(AdapterView<?> parent, View view, int position,
4            long id) {
5            String content = "植物名称:" + titles[position] + ".描述:"
6                + depicts[position];
7            Toast.makeText(MainActivity.this,
8                content, Toast.LENGTH_LONG).show();
9        }
10    });
```

上述代码中第 1 行使用 setOnItemClickListener()方法为 ListView 注册一个项被单击的监听器。当用户单击 ListView 中的任意 Item 的时候就会回调 onItemClick()方法,在这个方法中通过 position 参数判断用户单击的是哪一个子项,然后获取到相应数组对应的项。最后通过 Toast 将植物的信息展示出来。

代码运行效果如图 5-7 所示。

对于 ListView 中的每个 Item,当长按某一个 Item 的时候可以触发长按事件,下面以代码的形式说明如何使用长按事件。

在 onCreate()方法中添加以下代码。

程序清单 5-13: chart0503\java\com\example\administrator\chart0503\MainActivity.java

```
1    mListView.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
2        @Override
3        public boolean onItemLongClick(AdapterView<?>
4            parent, View view, int position, long id) {
5            AlertDialog alertDialog = new AlertDialog
6                .Builder(MainActivity.this).setMessage
7                ("确定要删除吗?")
```



视频详解

```

8          .setPositiveButton("确定", null)
9          .setNegativeButton("取消", null)
10         .create();
11         alertDialog.show();
12         return true;
13     }
14 });

```

在上述代码中第 1 行使用 `setOnItemLongClickListener()` 方法为 `ListView` 注册一个监听器。当用户长按 `ListView` 中的任意 `Item` 的时候就会回调 `onItemLongClick()` 方法,在这个方法中通过创建对话框,提示用户是否确定要删除。

第 5~13 行代码创建一个对话框,通过调用 `AlertDialog` 的静态内部类 `Builder` 的方法创建 `AlertDialog` 对象。通过调用 `Builder` 类的 `setMessage()` 方法设置提示的内容,通过 `setPositiveButton()` 和 `setNegativeButton()` 设置对话框的“确定”和“取消”按钮。通过调用 `AlertDialog` 对象的 `show()` 方法显示该对话框。

上述代码运行效果如图 5-8 所示。



图 5-7 程序运行效果



图 5-8 程序运行效果



视频详解

5.2.4 实战演练——订单中心

通过对 `ListView` 控件的学习,我们可以对“欢乐购商城”中的订单页面进行实现。布局页面代码如下。

程序清单 5-14: `chart0504\app\src\main\res\layout\activity_main.xml`

```

1 <?xml version = "1.0" encoding = "utf - 8"?>
2 <RelativeLayout xmlns:android = "http://schemas.android.com/apk/res/android"

```

```
3   xmlns:tools = "http://schemas.android.com/tools"
4       android:layout_width = "match_parent"
5       android:layout_height = "match_parent"
6       tools:context = ".MainActivity">
7       <ListView
8           android:id = "@ + id/order_ListView"
9           android:layout_width = "match_parent"
10          android:layout_height = "match_parent"/>
11   </RelativeLayout>
```

其中 Item 的布局文件如下。

程序清单 5-15: chart0504\ListView\res\layout\order_list_items.xml

```
1   <?xml version = "1.0" encoding = "utf-8"?>
2   <RelativeLayout xmlns:android = "http://schemas.android.com/apk/res/android"
3       android:layout_width = "match_parent"
4       android:layout_height = "match_parent">
5
6       <TextView
7           android:id = "@ + id/goods_title"
8           android:layout_width = "match_parent"
9           android:layout_height = "wrap_content"
10          android:layout_margin = "8dp"
11          android:text = "云南天然干花花束真花勿忘我家居客厅摆设超大花束"
12          android:textSize = "15sp"
13          android:textStyle = "bold" />
14
15       <TextView
16           android:id = "@ + id/order_status"
17           android:layout_width = "wrap_content"
18           android:layout_height = "wrap_content"
19           android:layout_below = "@ id/goods_title"
20           android:layout_marginLeft = "5dp"
21           android:text = "订单状态:已结算" />
22
23       <TextView
24           android:id = "@ + id/order_value"
25           android:layout_width = "wrap_content"
26           android:layout_height = "wrap_content"
27           android:layout_alignParentRight = "true"
28           android:layout_below = "@ id/goods_title"
29           android:layout_marginRight = "8dp"
30           android:text = "总价:80 元" />
31
32       <TextView
33           android:id = "@ + id/order_time"
34           android:layout_width = "wrap_content"
35           android:layout_height = "wrap_content"
```

```
38         android:layout_below = "@id/order_status"
39         android:layout_margin = "8dp"
40         android:text = "创建时间:" />
41     </RelativeLayout>
```

MainActivity 页面的代码如下。

程序清单 5-16: chart0504\java\com\example\administrator\chart0504\MainActivity.java

```
1     public class MainActivity extends AppCompatActivity {
2         private ListView orderList;
3         private String[] orderTitle = {"云南天然干花花束真花勿忘我家居客厅摆设超大花
4             束", "2019 秋装新款白色纯棉长袖 t 恤女装宽松春秋打底衫体恤大码上衣",
5             "鞋袋子装鞋子的收纳袋旅行鞋包收纳包防尘袋家用鞋罩束口鞋袋鞋套"};
6         private String[] orderValue = {"145", "213", "89"};
7         @Override
8         protected void onCreate(Bundle savedInstanceState) {
9             super.onCreate(savedInstanceState);
10            setContentView(R.layout.activity_main);
11            orderList = (ListView) findViewById(R.id.order_ListView);
12            orderList.setAdapter(new myListAdapter());
13        }
14        private class myListAdapter extends BaseAdapter {
15            private ViewHolder viewHolder;
16            @Override
17            public int getCount() {
18                return orderTitle.length;
19            }
20            @Override
21            public Object getItem(int position) {
22                return orderTitle[position];
23            }
24            @Override
25            public long getItemId(int position) {
26                return position;
27            }
28            @Override
29            public View getView(int position, View convertView, ViewGroup parent) {
30                if (convertView == null) {
31                    convertView = View.inflate(MainActivity.this,
32                        R.layout.order_list_items, null);
33                    viewHolder = new ViewHolder();
34                    viewHolder.orderTitle = (TextView) convertView.
35                        findViewById(R.id.goods_title);
36                    viewHolder.totalValue = (TextView)
37                        convertView.findViewById(R.id.order_value);
38                    viewHolder.orderTime = (TextView)
39                        convertView.findViewById(R.id.order_time);
40                    convertView.setTag(viewHolder);
41                }
42                viewHolder.orderTitle.setText(orderTitle[position]);
43                viewHolder.totalValue.setText(orderValue[position]);
44                viewHolder.orderTime.setText(String.valueOf(position));
45            }
46        }
47    }
```

```

36         } else {
37             viewHolder = (ViewHolder) convertView.getTag();
38         }
39         SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
40         viewHolder.orderTitle.setText(orderTitle[position]);
41         viewHolder.totalValue.setText("总价:" + orderValue[position] + "元");
42         viewHolder.orderTime.setText("创建时间:" + df.format(new Date()));
43         return convertView;
44     }
45
46     class ViewHolder {
47         TextView orderTitle;
48         TextView totalValue;
49         TextView orderTime;
50     }
51 }
52 }

```

上述程序运行结果如图 5-9 所示。

云南天然干花花束真花勿忘我家居客厅摆设超大花束	
订单状态: 已结算	总价: 145元
创建时间: 2020-01-31 03:21:23	
2019秋装新款白色纯棉长袖t恤女装宽松春秋打底衫 体恤大码上衣	
订单状态: 已结算	总价: 213元
创建时间: 2020-01-31 03:21:23	
鞋袋子装鞋子的收纳袋旅行鞋包收纳包防尘袋家用鞋 罩束口袋鞋套	
订单状态: 已结算	总价: 89元
创建时间: 2020-01-31 03:21:23	

图 5-9 程序运行效果



5.3 网格列表 GridView



任务陈述

“欢乐购商城”首页使用 GridView 控件使推荐商品按照网格的形式展示,下面对首页中的推荐商品列表做概要分析。



视频详解

分析：在首页中需要展示推荐商品，展示的商品只需要展示概要信息并尽可能多展示一些商品，如果采用列表的形式展示，每行只能展示一个列表项。为了多展示一些商品信息，在项目中采用 GridView 控件展示商品，使每行可展示两个列表项，如图 5-10 所示。



图 5-10 列表展示商品

相关知识



视频详解

5.3.1 GridView 控件

前面所学的 Spinner 和 ListView 显示列表数据时，都是以垂直方向显示，每行只显示一个列表项。无法实现一行显示多个列表项的效果。如果要实现多行多列的展示效果，可使用 GridView 控件。GridView 可将界面划分为若干个网格，可以设置每一行所能显示的列表项数量，然后根据总的列表项数来计算一共有多少行。例如，总共有 15 个列表项，每行存放 4 个列表项，则包含 4 行，如果每行存放 3 个列表项，则包含 5 行。使用 GridView 时，关键属性如表 5-2 所示。

表 5-2 GridView 关键属性

属 性 名	功 能 描 述
android: numColumns	每行中列的数量
android: horizontalSpacing	设置两个列表项之间的水平间距
android: verticalSpacing	设置两个列表项之间的垂直间距

下面以展示植物信息的例子讲解 GridView 的用法。

(1) 创建程序,创建名字为 chart0503 的应用程序。

(2) 导入图片,在 Android Studio 中切换到 Project 选项卡,选中程序中 res 文件夹,右击选择 New→Directory 选项,创建一个名为 drawable-hdpi 的文件夹。将程序所需要的图片导入到文件夹。

(3) 设置布局文件,在 res/layout/activity_main.xml 文件中添加一个 ListView 控件,并对相应属性进行设置。布局文件的核心代码如下。

程序清单 5-17: chart0505\app\src\main\res\layout\activity_main.xml

```
1 <GridView
2     android:id="@+id/gridView"
3     android:layout_width="match_parent"
4     android:layout_height="wrap_content"
5     android:horizontalSpacing="3dp"
6     android:numColumns="4"
7     android:verticalSpacing="3dp"></GridView>
```

(4) 设置每个 Item 的布局,在 res/layout 文件夹中创建一个 Item 界面的布局文件 grid_item.xml,在该文件中添加一个 ImageView 用来展示植物的图片,添加一个 TextView 用于展示植物的名称。完整的布局文件代码如下。

程序清单 5-18: chart0505\app\src\main\res\layout\grid_item.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="wrap_content"
4     android:layout_height="wrap_content"
5     android:background="#cccccc"
6     android:gravity="center_horizontal"
7     android:orientation="vertical"
8     android:padding="3dp">
9     <ImageView
10         android:id="@+id/image"
11         android:layout_width="100dp"
12         android:layout_height="90dp"
13         android:scaleType="fitXY"
14         android:src="@drawable/a" />
15     <TextView
16         android:id="@+id/tv_title"
17         android:layout_width="wrap_content"
18         android:layout_height="wrap_content"
19         android:layout_below="@id/image"
20         android:text="铁茉莉"
21         android:textSize="15sp" />
22 </LinearLayout>
```

(5) 在 MainActivity 中创建一个 SimpleAdapter 对象,gridView 对象通过 setAdapter() 方法将 SimpleAdapter 对象与 gridView 控件关联起来,实现数据适配。完整的代码如下。

程序清单 5-19: chart0505\java\com\example\administrator\chart0505\MainActivity.java

```
1 public class MainActivity extends AppCompatActivity {
2     private GridView gridView;
3     private List<Map<String, Object>> datas = new ArrayList<Map<String, Object>>();
4
5     private int[] icons = {R.drawable.a, R.drawable.b, R.drawable.c,
6         R.drawable.d, R.drawable.e, R.drawable.f, R.drawable.g,
7         R.drawable.h, R.drawable.i, R.drawable.j, R.drawable.k,
8         R.drawable.l};
9     private String[] titles = {"白鹃梅", "五色椒", "小檗",
10         "李叶绣菊", "多花兰", "獐耳细辛", "铁茉莉", "三色狸藻", "东洋菊",
11         "大花皇冠", "苹果花", "三弄芙蓉"};
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16         gridView = (GridView) findViewById(R.id.gridView);
17         initData();
18         SimpleAdapter simpleAdapter = new SimpleAdapter
19             (this, datas, R.layout.grid_item, new
20             String[]{"image", "titles"}, new int[]{R
21                 .id.image, R.id.tv_title});
22         gridView.setAdapter(simpleAdapter);
23     }
24     private void initData() {
25         for (int i = 0; i < titles.length; i++) {
26             Map<String, Object> item = new
27                 HashMap<String, Object>();
28             item.put("titles", titles[i]);
29             item.put("image", icons[i]);
30             datas.add(item);
31         }
32     }
33 }
```

上述代码中,在第3行定义一个 List 集合对象,用于存储植物标题和植物图片的信息。第5行和第9行分别定义两个数组用于保存植物的标题和图片。

第24~32行定义一个方法 initData()将数组中的数据以键值对的形式保存在 Map 对象中,将每个 Map 对象保存到 List 集合对象中。

第18行定义 SimpleAdapter 对象,调用其构造函数,并传入5个相应的参数。第一个参数表示上下文对象,第二个参数表示数据源,第三个参数表示 Item 布局资源的 id,第四个参数表示 Map 集合中的 key 值,第五个参数表示 Item 布局中 key 值所代表的 value 所绑定的控件。

第22行通过 setAdapter()将 SimpleAdapter 对象设置给 gridView 实现数据适配。

上述程序运行效果如图 5-11 所示。

在上述案例中,GridView 控件使用的是 SimpleAdapter 对象实现了数据的适配,如果

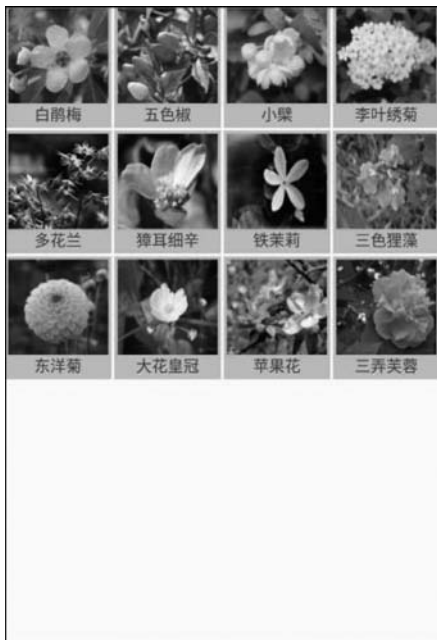


图 5-11 程序运行效果

Item 要实现更加复杂的布局,可采用继承 BaseAdapter 抽象类的方法来实现。BaseAdapter 的使用已在 ListView 中做过相应展示,在这里不再演示。对于 GridView 中每个 Item 也可以设置其单击事件,包括普通的单击事件和长按事件,其用法已在 ListView 中做过相应的展示,在这里不再演示。

5.3.2 实战演练——首页商品列表

通过对 GridView 的学习,我们可以对“欢乐购商城”中的首页进行实现。其中布局页面的代码如下。

程序清单 5-20: chart0506\app\src\main\res\layout\activity_main.xml

```
1  <?xml version = "1.0" encoding = "utf-8"?>
2  <ScrollView xmlns:android = "http://schemas.android.com/apk/res/android"
3      android:layout_width = "match_parent"
4      android:layout_height = "match_parent"
5      android:gravity = "center">
6      <LinearLayout
7          android:layout_width = "match_parent"
8          android:layout_height = "match_parent"
9          android:orientation = "vertical">
10         <ImageView
11             android:layout_width = "match_parent"
12             android:layout_height = "150dp"
13             android:scaleType = "fitXY"
14             android:src = "@drawable/jhs" />
```



视频详解

```
15         <TableLayout
16             android:id="@+id/table_smallIcon"
17             android:layout_width="match_parent"
18             android:layout_height="wrap_content"
19             android:gravity="center">
20         <TableRow>
21             <ImageView
22                 android:id="@+id/a"
23                 android:layout_width="73dp"
24                 android:layout_height="73dp"
25                 android:clickable="true"
26                 android:src="@drawable/a" />
27             <ImageView
28                 android:id="@+id/b"
29                 android:layout_width="73dp"
32                 android:layout_height="73dp"
33                 android:src="@drawable/b" />
34             <ImageView
35                 android:id="@+id/c"
36                 android:layout_width="73dp"
37                 android:layout_height="73dp"
38                 android:src="@drawable/c" />
39             <ImageView
40                 android:id="@+id/d"
41                 android:layout_width="73dp"
42                 android:layout_height="73dp"
43                 android:src="@drawable/d" />
44             <ImageView
45                 android:id="@+id/e"
46                 android:layout_width="73dp"
47                 android:layout_height="73dp"
48                 android:src="@drawable/e" />
49         </TableRow>
50         <TableRow>
51             <ImageView
52                 android:id="@+id/f"
53                 android:layout_width="73dp"
54                 android:layout_height="73dp"
55                 android:src="@drawable/f" />
56             <ImageView
57                 android:id="@+id/g"
58                 android:layout_width="73dp"
59                 android:layout_height="73dp"
60                 android:src="@drawable/g" />
61             <ImageView
62                 android:id="@+id/h"
63                 android:layout_width="73dp"
64                 android:layout_height="73dp"
65                 android:src="@drawable/h" />
```

```

66         < ImageView
67             android:id = "@ + id/i"
68             android:layout_width = "73dp"
69             android:layout_height = "73dp"
70             android:src = "@drawable/i" />
71         < ImageView
72             android:id = "@ + id/k"
73             android:layout_width = "73dp"
74             android:layout_height = "73dp"
75             android:src = "@drawable/k" />
76     </TableRow>
77 </TableLayout>
78 < com.example.administrator.taobao.adapter.MyGridView
79     android:id = "@ + id/gridView"
80     android:layout_width = "match_parent"
81     android:layout_height = "wrap_content"
82     android:layout_marginTop = "5dp"
83     android:gravity = "center"
84     android:horizontalSpacing = "5dp"
85     android:numColumns = "2"
86     android:verticalSpacing = "5dp">
87 </com.example.administrator.taobao.adapter.MyGridView>
88 </LinearLayout>
89 </ScrollView>

```

其中,Item 的布局文件如下。

程序清单 5-21: chart0506\app\src\main\res\layout\goods_detail_item.xml

```

1  <?xml version = "1.0" encoding = "utf - 8"?>
2  < RelativeLayout xmlns:android = "http://schemas.android.com/apk/res/android"
3      android:layout_width = "wrap_content"
4      android:layout_height = "wrap_content">
5      < ImageView
6          android:id = "@ + id/iv_iamege"
7          android:layout_width = "160dp"
8          android:layout_height = "160dp"
9          android:layout_marginTop = "4dp" />
10     < TextView
11         android:id = "@ + id/tv_title"
12         android:layout_width = "160dp"
13         android:layout_height = "wrap_content"
14         android:layout_below = "@ + id/iv_iamege"
15         android:ellipsize = "end"
16         android:maxLines = "1"
17         android:text = "标题测试数据"
18         android:textSize = "14dp" />
19     < TextView
20         android:id = "@ + id/tv_prices"

```

```
21         android:layout_width = "wrap_content"
22         android:layout_height = "wrap_content"
23         android:layout_below = "@id/tv_title"
24         android:text = "¥ 10"
25         android:textColor = "#f00"
26         android:textSize = "16sp" />
27     <TextView
28         android:id = "@ + id/tv_numbers"
29         android:layout_width = "wrap_content"
32         android:layout_height = "wrap_content"
33         android:layout_alignBottom = "@id/tv_prices"
34         android:layout_marginLeft = "10dp"
35         android:layout_toRightOf = "@id/tv_prices"
36         android:text = "1000 人付款" />
37 </RelativeLayout>
```

定义一个适配器类,代码如下。

程序清单 5-22: chart0506\java\com\example\administrator\chart0506\GoodsAdapter. Java

```
1 public class GoodsAdapter extends BaseAdapter {
2     private Context context;
3     private List<Good> goodsList;
4     private ViewHolder viewHolder;
5     public GoodsAdapter(Context context, List<Good> goodsList) {
6         this.context = context;
7         this.goodsList = goodsList;
8     }
9     @Override
10    public int getCount() {
11        return goodsList.size();
12    }
13    @Override
14    public Object getItem(int position) {
15        return goodsList.get(position);
16    }
17    @Override
18    public long getItemId(int position) {
19        return position;
20    }
21    @Override
22    public View getView(int position, View convertView, ViewGroup parent) {
23        if (convertView == null) {
24            convertView = LayoutInflater.from(context).
25                inflate(R.layout.goods_detail_item, parent, false);
26            viewHolder = new ViewHolder();
27            viewHolder.image = convertView.findViewById(R.id.iv_image);
28            viewHolder.textTitle = convertView.findViewById(R.id.tv_title);
29            viewHolder.textPrice = convertView.findViewById(R.id.tv_prices);
```

```

29         viewHolder.textNumber = convertView.findViewById(R.id.tv_numbers);
30         convertView.setTag(viewHolder);
31     } else {
32         viewHolder = (ViewHolder) convertView.getTag();
33     }
34     viewHolder.image.setBackgroundResource(goodsList.get(position).getImage());
35     viewHolder.textTitle.setText(goodsList.get(position).getName());
36     viewHolder.textPrice.setText("¥ " + goodsList.get(position).getPrice());
37     viewHolder.textNumber.setText(goodsList.get(position).getVolume() + "人付款");
38     return convertView;
39 }
40 class ViewHolder {
41     ImageView image;
42     TextView textTitle;
43     TextView textPrice;
44     TextView textNumber;
45 }
46 }

```

MainActivity 页面的代码如下。

程序清单 5-23: chart0506\java\com\example\administrator\chart0506\MainActivity.java

```

1  public class MainActivity extends AppCompatActivity {
2
3      private MyGridView gridView;
4      private List<Good> goodList = new ArrayList<Good>();
5      private int[] picttrue = {R.drawable.a000, R.drawable.a001, R.drawable.a002,
6          R.drawable.a003, R.drawable.a004, R.drawable.a005, R.drawable.a006,
7          R.drawable.a007, R.drawable.a008, R.drawable.a009};
8      private String[] title = {"2019 秋装新款白色纯棉长袖 t 恤女装宽松春秋打底衫体恤
9          大码上衣",
10         "2019 初秋新款潮韩版洋气女装宽松秋季短款衬衫女秋装方领长袖上衣",
11         "纯棉长袖 T 恤女上衣 2019 新款竹节棉宽松秋衣女装薄款初秋打底衫潮",
12         "半高领打底衫女装秋冬薄款洋气莫代尔堆堆领长袖 t 恤内搭针织上衣",
13         "女装 2019 新款潮超火 cec 短袖女春夏季港味宽松百搭上衣 T 恤 ins 洋气",
14         "夏大大胖 mm 韩版宽松秋装新款女装大码 2019 牛仔套装外套减龄连衣裙",
15         "大姗姗家瘦瘦裤胖 mm 韩版大码女装秋季百搭外穿打底裤网红款小脚裤",
16         "中国大陆", '夏装紧身纯白色 T 恤女短袖修身女装女士纯棉 2019 新款上衣潮打底衫',
17         "纯棉长袖 T 恤女上衣 2019 新款竹节棉宽松秋衣女装薄款初秋打底衫潮",
18         "纯棉红色 v 领 T 恤女短袖修身夏女装 2019 新款潮体恤紧身上衣黑色短款", };
19     private String[] prices = {"39.9", "29.87", "29", "56", "20", "138", "97", "15",
20         "78", "30"};
21     private String[] number = {"87816", "37384", "62165", "43221", "4561", "43211",
22         "3011", "764", "7656", "4567"};
23     @Override

```

```
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_main);
20         initView();
21     }
22     private void initView() {
23         for (int i = 0; i < 10; i++) {
24             Good good = new Good();
25             good.setImage(pictrue[i]);
26             good.setName(title[i]);
27             good.setPrice(Double.valueOf(prices[i]));
28             good.setVolume(number[i]);
29             goodList.add(good);
30         }
31         gridView = (MyGridView) findViewById(R.id.gridView);
32         gridView.setAdapter(new GoodsAdapter(MainActivity.this, goodList));
33         gridView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
34             @Override
35             public void onItemClick(AdapterView<?> parent, View view, int position,
36                 long id) {
37             }
38         });
39     }
40 }
```

上述程序运行效果如图 5-12 所示。



图 5-12 程序运行效果



5.4 增强列表 RecyclerView



视频详解



任务陈述

“欢乐购商城”项目中商品列表页面使用了 RecyclerView 控件,下面对商品列表页面做概要分析。

分析: 在商品列表页面中,需要展示分类商品的概要信息,包括商品图片、商品标题、商品产地和商品所属店铺等。每个列表项的布局较复杂,在项目中采用 RecyclerView 控件实现商品列表展示功能,如图 5-13 所示。



图 5-13 商品列表展示



相关知识

5.4.1 RecyclerView 控件

RecyclerView 是 Android 5.0 以后提供的更为强大的列表控件,不仅可以轻松实现 ListView、GridView 的效果,还优化了之前列表控件存在的各种不足之处,是目前官方推荐的列表控件。

RecyclerView 控件可通过 LayoutManager 类设置列表项的布局管理器,用于控制列表项的整体布局,常见的布局管理器有: LinearLayoutManager (线性布局管理器,可以控制列表项按照水平从左到右摆放,或者垂直从上到下摆放)、GridLayoutManager (网格布局管



视频详解

理器,按照若干行或者若干列来摆放列表项)、StaggeredGridLayoutManager (交错网格布局管理器,可以实现瀑布流效果,网格不是整齐的而是有所偏移)。

RecyclerView 控件使用 RecyclerView.Adapter 适配器,该适配器将 BaseAdapter 中的 getView()方法拆分为 onCreateViewHolder()方法和 onBindViewHolder()方法,强制使用 ViewHolder 类,使代码编写更加规范,提高性能。RecyclerView 控件复用 Item 对象是由控件自身完成,提高了灵活性。

为了使 RecyclerView 在所有的 Android 版本上都可以使用,Android 团队将其定义在兼容包中,因此如果想使用 RecyclerView 控件首先需要在项目的 build.gradle 中添加相关的依赖库。其中,RecyclerView 的版本可根据本机上安装的 Android 版本变化。

接下来通过加载植物信息的例子讲解 RecyclerView 的使用,并实现垂直列表、水平列表和网格列表效果。

(1) 创建程序,创建名字为 chart0507 的应用程序。

(2) 导入图片,在 Android Studio 中切换到 Project 选项卡,选中程序中 res 文件夹,右击,选择 New→Directory 选项,创建一个名为 drawable-hdpi 的文件夹,将程序所需要的图片导入文件夹。

(3) RecyclerView 是 Android 5.0 新增的控件,为了让 RecyclerView 在所有的 Android 版本中都能使用,RecyclerView 被定义在 support 库中,因此在使用时,需要在项目的 build.gradle 中添加相应的依赖库。打开 app\build.gradle 文件,在 dependencies{} 节点中添加 com.android.support:recyclerview-v7 库。

```
dependencies {  
    implementation fileTree(include: ['*.jar'], dir: 'libs')  
    implementation 'com.android.support:appcompat-v7:28.0.0'  
    implementation 'com.android.support:recyclerview-v7:28.0.0'  
    ...  
}
```

注意: 添加的 com.android.support:recyclerview-v7 库的版本需要与 com.android.support:appcompat 库的版本一致,否则会报错。

(4) 设置布局文件,在 res\layout\activity_main.xml 文件中添加一个 RecyclerView 控件,并设置相应属性,布局文件的核心代码如下。

程序清单 5-24: char0507\app\src\main\res\layout\activity_main.xml

```
1 <android.support.v7.widget.RecyclerView  
2     android:id="@+id/recyclerView"  
3     android:layout_width="match_parent"  
4     android:layout_height="match_parent">  
5 </android.support.v7.widget.RecyclerView>
```

(5) 设置每个 Item 的布局,由于每个 Item 的布局效果和 GridView 中的显示效果一样,把上一个项目的图片复制过来,新建一个布局文件命名为 grid_item.xml。把上一节例子中的 Item 布局复制过来。

(6) 在 MainActivity 中获取 RecyclerView 对象并进行数据适配,最终将数据显示在列表界面中。完整的代码如下。

程序清单 5-25: chart0507\java\com\example\administrator\chart0507\MainActivity.java

```
1 public class MainActivity extends AppCompatActivity {
2
3     private int[] icons = {R.drawable.a, R.drawable.b, R
4         .drawable.c, R.drawable.d, R.drawable.e,
5         R.drawable.f, R.drawable.g, R.drawable.h, R
6         .drawable.i, R
7         .drawable.j, R.drawable.k, R.drawable.l};
8     private String[] titles = {"白鹃梅", "五色椒", "小檗",
9         "李叶绣菊", "多花兰", "獐耳细辛", "铁茉莉", "三色狸藻", "东洋菊",
10        "大花皇冠", "苹果花", "三弄芙蓉"};
11     private RecyclerView recyclerView;
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16         recyclerView = (RecyclerView) findViewById(R.id
17             .recyclerView);
18         LinearLayoutManager layoutManager = new
19             LinearLayoutManager(MainActivity.this);
20         recyclerView.setLayoutManager(layoutManager);
21         MyAdapter myAdapter = new MyAdapter();
22         recyclerView.setAdapter(myAdapter);
23     }
24     public class MyAdapter extends RecyclerView
25         .Adapter<MyAdapter.MyViewHolder> {
26         @Override
27         public MyViewHolder onCreateViewHolder(ViewGroup viewGroup, int viewType) {
28             MyViewHolder myViewHolder = new MyViewHolder
29                 (LayoutInflater.from(MainActivity.this)
30                 .inflate(R.layout.recycler_item, viewGroup, false));
31             return myViewHolder;
32         }
33         @Override
34         public void onBindViewHolder(MyViewHolder myViewHolder, int position) {
35             myViewHolder.image.setImageResource
36                 (icons[position]);
37             myViewHolder.titleView.setText
38                 (titles[position]);
39         }
40         @Override
41         public int getItemCount() {
42             return titles.length;
43         }
44     }
```

```

43         public class MyViewHolder extends RecyclerView
44             .ViewHolder {
45             ImageView image;
46             TextView titleView;
47             public MyViewHolder(View itemView) { super(itemView);
48                 image = (ImageView) itemView.findViewById(R.id.image);
49                 titleView = (TextView) itemView.findViewById(R.id.tv_title);
50             }
51         }
52     }
53 }

```

上述代码中,在第3~10行定义两个数组对象,用于存储植物标题和植物图片的信息。第16~22行首先获取 recyclerView 对象并为 recyclerView 创建布局管理器,其中 layoutManager 对象用于指明 recyclerView 为线性布局方式,默认为垂直方向的线性布局。通过 setLayoutManager() 将 layoutManager 对象设置给 recyclerView 对象。之后创建 MyAdapter 对象,最后通过 setAdapter() 方法将适配器 MyAdapter 对象设置到 recyclerView 控件上,这样 recyclerView 就完成了数据的适配。

第24~42行为 recyclerView 创建适配器对象,新建一个内部类 MyAdapter 继承自 RecyclerView.Adapter 并将 ViewHolder 泛型参数指定为 MyAdapter.MyViewHolder,其中 MyViewHolder 是 MyAdapter 中定义的一个内部类。此外,RecyclerView.Adapter 类是一个抽象类,还需要实现里面的三个抽象方法: onCreateViewHolder()、onBindViewHolder()、getItemCount()。onCreateViewHolder() 中创建 MyViewHolder 对象,并将 recycler_item 布局文件转换成对象之后传入到 MyViewHolder 构造函数中,最后将 myViewHolder 对象返回。onBindViewHolder() 用于对 recyclerView 每个 Item 项的数据进行赋值,会在每个子项滚动到屏幕内的时候执行,并通过 position 获取 Item 的索引。getItemCount() 是获取列表总条目数。

第43~51行定义 MyViewHolder 类继承自 RecyclerView.ViewHolder。在该类中获取 Item 界面的控件,在 MyViewHolder 的构造函数中需要传入一个参数 itemView,这个参数通常是 recyclerView 中 Item 项的最外层布局,对于 itemView 对象通过 findViewById() 获取到 Item 布局中的每一个控件对象。

上述程序运行效果如图 5-14 所示。

5.4.2 RecyclerView 实现横向和网格布局

在使用 RecyclerView 控件时需要注意在运行之前一定要为 RecyclerView 指定列表项的布局管理



图 5-14 程序运行效果



视频详解

器,否则系统不知道该如何显示,就会报错并强制退出。另外,RecyclerView 不仅可以实现纵向布局,还可以实现横向布局。实现横向布局的主要代码如下。

```
LinearLayoutManager linearLayoutManager = newLinearLayoutManager(MainActivity.this,  
    LinearLayoutManager.HORIZONTAL, false);  
recyclerView.setLayoutManager(linearLayoutManager);
```

上述代码中在创建线性布局管理器时传递了三个参数,第一个参数为上下文对象,通常为当前的 Activity;第二个参数用于指定方向,默认为垂直的,在此指定为水平的;第三个参数表示列表项的顺序是否反转,对于水平方向来说,列表项默认是从左到右摆放,如果需从右到左摆放,则表示需要反转,传递 true 进去即可。

除了 LinearLayoutManager 之外,RecyclerView 还提供了 GridLayoutManager 和 StaggeredGridLayoutManager 这两种内置的布局排列方式。GridLayoutManager 可以用于实现网格布局,StaggeredGridLayoutManager 可以用于实现瀑布流布局。如果要实现网格布局,代码如下。

```
GridLayoutManager gridLayoutManager = newGridLayoutManager  
(this, 4, GridLayoutManager.VERTICAL, false);  
recyclerView.setLayoutManager(gridLayoutManager);
```

创建网格布局管理器时需传递四个参数,第一个参数为上下文对象;第二个参数为每行或者每列排列 Item 个数;第三个参数表示网格的方向,是水平摆放还是垂直摆放,GridLayoutManager.VERTICAL 表示纵向排列;第四个参数表示列表项的顺序是否反转,false 表示不反转,按照原有次序加载。上述代码运行结果如图 5-15 所示。

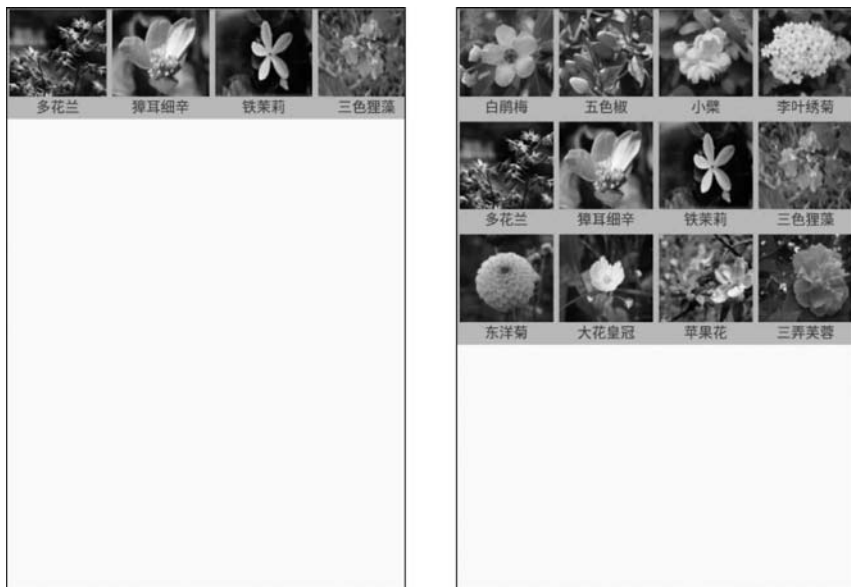


图 5-15 程序运行效果



视频详解

5.4.3 RecyclerView 实现单击事件

在 ListView 和 GridView 中对于每个 Item 都有单击事件,对于 RecyclerView 同样也有单击事件,不过不同于 ListView,RecyclerView 并没有提供类似于 setOnItemClickListener() 这样的注册监听器方法,需要我们给 Item 中具体的 View 去注册单击事件,看似编写代码变复杂,其实是增加了控件的灵活性,这也是比 ListView 更强大的地方之一。对于 RecyclerView 控件的单击事件,需要修改自定义类 MyAdapter 中的代码,单击事件是定义在 onCreateViewHolder() 方法中,具体代码如下。

程序清单 5-26: chart0507\java\com\example\administrator\chart0507\MainActivity.java

```
1 public class MyAdapter extends RecyclerView.Adapter<MyAdapter.MyViewHolder> {
2     @Override
3     public MyViewHolder onCreateViewHolder(ViewGroup viewGroup, int viewType) {
4         final MyViewHolder myViewHolder = new
5         MyViewHolder (LayoutInflater.from(MainActivity
6         .this).inflate(R.layout.recycler_item, viewGroup, false));
7         myViewHolder.view.setOnClickListener(new View
8         .OnClickListener() {
9             @Override
10            public void onClick(View v) {
11                int adapterPosition = myViewHolder .getAdapterPosition();
12                Toast.makeText (MainActivity.this, "你单击的植物是:" + titles
13                [adapterPosition],
14                Toast.LENGTH_SHORT).show();
15            }
16        });
17        myViewHolder.image.setOnClickListener(new View.OnClickListener() {
18            @Override
19            public void onClick(View v) {
20                int adapterPosition = myViewHolder
21                .getAdapterPosition();
22                Toast.makeText(MainActivity.this,
23                "你单击图片的植物名是:" + titles[adapterPosition],
24                Toast.LENGTH_SHORT).show();
25            }
26        });
27        return myViewHolder;
28    }
29    @Override
30    public void onBindViewHolder(MyViewHolder myViewHolder, int position) {
31        myViewHolder.image.setImageResource (icons[position]);
32        myViewHolder.titleView.setText
33        (titles[position]);
34    }
35    @Override
36    public int getItemCount() {
37        return titles.length;
38    }
39 }
```

```
37     }
38     public class MyViewHolder extends RecyclerView.ViewHolder {
39         ImageView image;
40         TextView titleView;
41         View view;
42         public MyViewHolder(View itemView) {
43             super(itemView);
44             view = itemView;
45             image = (ImageView) itemView.findViewById
46                 (R.id.image);
47             titleView = (TextView) itemView
48                 .findViewById(R.id.tv_title);
49         }
50     }
51 }
```

上述代码中,第6行声明 view 变量保存 Item 最外层布局对象。第7~15行代码为最外层布局对象 view 添加单击事件,第11行获取用户单击的 position,第12~13行代码通过 Toast 弹出用户单击植物的名称。第16~27行为 ImageView 控件绑定单击事件,当用户单击 Item 项中的图片,通过 Toast 弹出用户单击图片所对应的植物的名称。

上述程序运行效果如图 5-16 所示。

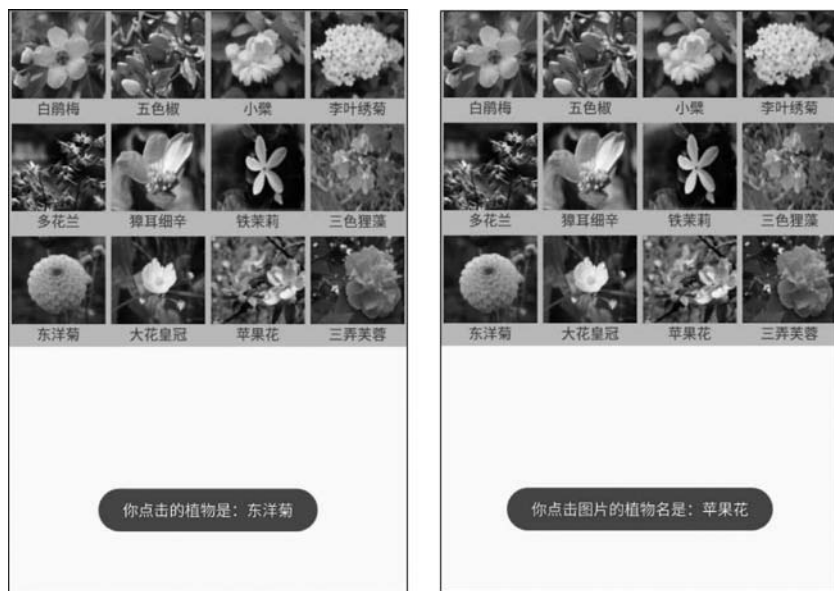


图 5-16 程序运行效果

5.4.4 实战演练——产品列表

通过对 RecyclerView 的学习,我们可以对“欢乐购商城”中商品列表页面进行实现。其布局页面的代码如下。



视频详解

程序清单 5-27: chart0508\app\src\main\res\layout\activity_main.xml

```
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:orientation="vertical"
6   tools:context=".MainActivity">
7
8     <android.support.v7.widget.RecyclerView
9       android:id="@+id/recycle_views"
10      android:layout_width="match_parent"
11      android:layout_height="match_parent">
12   </android.support.v7.widget.RecyclerView>
13 </LinearLayout>
```

其中 Item 的布局文件如下。

程序清单 5-28: chart0508\app\src\main\res\layout\goods_list_items.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent">
5   <ImageView
6     android:id="@+id/iv_iamge"
7     android:layout_width="120dp"
8     android:layout_height="120dp"
9     android:layout_marginTop="4dp"
10    android:src="@drawable/a000" />
11   <TextView
12     android:id="@+id/tv_title"
13     android:layout_width="match_parent"
14     android:layout_height="wrap_content"
15     android:layout_alignTop="@+id/iv_iamge"
16     android:layout_marginLeft="5dp"
17     android:layout_toRightOf="@id/iv_iamge"
18     android:ellipsize="end"
19     android:maxLines="2"
20     android:text="测试测试测试测试测试测试"
21     android:textSize="13dp" />
22   <TextView
23     android:id="@+id/tv_count"
24     android:layout_width="wrap_content"
25     android:layout_height="wrap_content"
26     android:layout_below="@id/tv_title"
27     android:layout_margin="10dp"
28     android:layout_toRightOf="@+id/iv_iamge"
29     android:text="1100" />
32  <TextView
```

```
33         android:layout_width = "wrap_content"
34         android:layout_height = "wrap_content"
35         android:layout_alignBottom = "@ + id/tv_count"
36         android:layout_marginRight = "3dp"
37         android:layout_toLeftOf = "@ + id/tv_address"
38         android:text = "产地:" />
39     <TextView
40         android:id = "@ + id/tv_address"
41         android:layout_width = "wrap_content"
42         android:layout_height = "wrap_content"
43         android:layout_alignBottom = "@ + id/tv_count"
44         android:layout_alignParentRight = "true"
45         android:text = "河南 商丘" />
46     <TextView
47         android:id = "@ + id/price_title"
48         android:layout_width = "wrap_content"
49         android:layout_height = "wrap_content"
50         android:layout_alignLeft = "@ + id/tv_count"
51         android:layout_below = "@ + id/tv_count"
52         android:layout_marginTop = "8dp"
53         android:text = "价格:¥" />
54     <TextView
55         android:id = "@ + id/tv_new_price"
56         android:layout_width = "wrap_content"
57         android:layout_height = "wrap_content"
58         android:layout_alignBottom = "@ + id/price_title"
59         android:layout_toRightOf = "@ + id/price_title"
60         android:text = "12.00"
61         android:textSize = "16sp"
62         android:textStyle = "bold" />
63     <LinearLayout
64         android:layout_width = "match_parent"
65         android:layout_height = "25dp"
66         android:layout_alignBottom = "@ + id/iv_image"
67         android:layout_alignLeft = "@ + id/price_title"
68         android:background = "@drawable/textview_border_style"
69         android:gravity = "center"
70         android:orientation = "horizontal">
71         <TextView
72             android:layout_width = "wrap_content"
73             android:layout_height = "wrap_content"
74             android:text = "店铺:" />
75         <TextView
76             android:id = "@ + id/tv_shopname"
77             android:layout_width = "wrap_content"
78             android:layout_height = "wrap_content"
79             android:text = "小飞的店铺"
80             android:textColor = "@color/colorPrimary" />
81     </LinearLayout>
82 </View>
```

```
83         android:id="@+id/view"
84         android:layout_width="match_parent"
85         android:layout_height="1dp"
86         android:layout_below="@id/iv_image"
87         android:layout_marginTop="3dp"
88         android:background="@color/grey" />
89     </RelativeLayout>
```

定义一个适配器类,代码如下。

程序清单 5-29: chart0508\app\src\main\java\com\example\administrator\goodsListAdapter

```
1     public class goodsListAdapter extends RecyclerView.Adapter < goodsListAdapter.
    ViewHolder> {
2         private List<Good> mList;
3         private Activity mActivity;
4         private Good good;
5         public goodsListAdapter(Activity activity, List<Good> list) {
6             mList = list;
7             mActivity = activity;
8         }
9         @Override
10        public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
11            final ViewHolder viewHolder = new ViewHolder(View.inflate(mActivity,
12                R.layout.goods_list_items, null));
13            viewHolder.itemView.setOnClickListener(new View.OnClickListener() {
14                @Override
15                public void onClick(View v) {
16                    int adapterPosition = viewHolder.getAdapterPosition();
17                    Intent intent = new Intent(mActivity, GoodDetailActivity.class);
18                    intent.putExtra("goodInfo", mList.get(adapterPosition));
19                    mActivity.startActivity(intent);
20                }
21            });
22            return viewHolder;
23        }
24        @Override
25        public void onBindViewHolder(ViewHolder holder, int position) {
26            good = mList.get(position);
27            holder.imageView.setImageResource(good.getImage());
28            holder.title.setText(good.getName());
29            holder.price.setText("" + good.getPrice());
30            holder.count.setText(good.getVolume() + "人付款");
31            holder.address.setText(good.getProductLocation());
32            holder.shopName.setText(good.getShopName());
33        }
34    }
35    @Override
36    public int getItemCount() {
```

```

37         return mList.size();
38     }
39     class ViewHolder extends RecyclerView.ViewHolder {
40         TextView shopName;
41         TextView title;
42         TextView count;
43         TextView address;
44         TextView price;
45         ImageView imageView;
46         View functionView;
47         public ViewHolder(View itemView) {
48             super(itemView);
49             functionView = itemView;
50             imageView = itemView.findViewById(R.id.iv_image);
51             title = (TextView) itemView.findViewById(R.id.tv_title);
52             count = (TextView) itemView.findViewById(R.id.tv_count);
53             address = (TextView) itemView.findViewById(R.id.tv_address);
54             price = (TextView) itemView.findViewById(R.id.tv_new_price);
55             shopName = (TextView) itemView.findViewById(R.id.tv_shopname);
56         }
57     }
58 }
59 }

```

MainActivity 页面的代码如下。

程序清单 5-30: chart0508\app\src\main\java\com\example\administrator\MainActivity.java

```

1     public class MainActivity extends AppCompatActivity {
2
3         private RecyclerView recyclerView;
4         private LinearLayoutManager layoutManager;
5         private List<Good> goodList = new ArrayList<Good>();
6         private int[] picture = {R.drawable.a000, R.drawable.a001, R.drawable.a002, R.
7         drawable.a003,R. drawable.a004, R.drawable.a005, R.drawable.a006,
8             R.drawable.a007, S. R.drawable.a008, R.drawable.a009};
9         private String[] title = {"2019 秋装新款白色纯棉长袖 t 恤女装宽松春秋打底衫体恤
10             大码上衣",
11             "2019 初秋新款潮韩版洋气女装宽松秋季短款衬衫女秋装方领长袖上衣",
12             "纯棉长袖 T 恤女上衣 2019 新款竹节棉宽松秋衣女装薄款初秋打底衫潮",
13             "半高领打底衫女装秋冬薄款洋气莫代尔堆堆领长袖 t 恤内搭针织上衣",
14             "女装 2019 新款潮超火 cec 短袖女春夏季港味宽松百搭上衣 T 恤 ins 洋气",
15             "夏大大胖 mm 韩版宽松秋装新款女装大码 2019 牛仔套装外套减龄连衣裙",

```

```

11         "大姗姗家瘦瘦裤胖 mm 韩版大码女装秋季百搭外穿打底裤网红款小脚裤",
12         "中国大陆", "夏装紧身纯白色 T 恤女短袖修身女装女士纯棉 2019 新款上衣潮打底衫",
13         "纯棉长袖 T 恤女上衣 2019 新款竹节棉宽松秋衣女装薄款初秋打底衫潮",
14         "纯棉红色 v 领 T 恤女短袖修身夏女装 2019 新款潮体恤紧身上衣黑色短款", };
15     private String[] location = {"云南 昆明", "广东 广州", "江西 南昌", "广东 汕头",
16         "浙江 金华", "广东 广州", "江苏 淮安",
17         "江苏 苏州", "广东 深圳", "安徽 芜湖"};
18     private String[] shop = {"lovtis 兰蒂斯旗舰店", "lovtis 兰蒂斯旗舰店",
19         "lovtis 兰蒂斯旗舰店", "lovtis 兰蒂斯旗舰店",
20         "旗舰店", "旗舰店", "旗舰店", "旗舰店", "旗舰店", "旗舰店"};
21     private String[] number = {"87816", "37384", "62165", "43221", "4561", "43211",
22         "3011", "764", "7656", "4567"};
23     private String[] prices = {"39.9", "29.87", "29", "56", "20", "138", "97", "15",
24         "78", "30"};
25     @Override
26     protected void onCreate(Bundle savedInstanceState) {
27         super.onCreate(savedInstanceState);
28         setContentView(R.layout.activity_main);
29         initView();
30     }
31     private void initView() {
32         for (int i = 0; i < 10; i++) {
33             Good good = new Good();
34             good.setImage(pictrue[i]);
35             good.setName(title[i]);
36             good.setProductLocation(location[i]);
37             good.setShopName(shop[i]);
38             good.setPrice(Double.valueOf(prices[i]));
39             good.setVolume(number[i]);
40             goodList.add(good);
41         }
42         recyclerView = (RecyclerView) findViewById(R.id.recycle_Views);
43         linearLayoutManager = new LinearLayoutManager(this);
44         recyclerView.setLayoutManager(linearLayoutManager);
45         recyclerView.setItemAnimator(new DefaultItemAnimator());
46         recyclerView.addItemDecoration ( new DividerItemDecoration
47             (this, DividerItemDecoration
48                 .VERTICAL));
49         recyclerView.setAdapter(new goodsListAdapter(this, goodList));
50     }
51 }

```

上述程序运行效果如图 5-17 所示。



图 5-17 程序运行效果

本章小结

本章围绕“欢乐购商城”项目首页、产品列表页面、订单列表页面和修改收货地址页面引入 Android 中列表控件的讲解。详细介绍了常用的列表控件的使用场景和基本属性,并通过实现首页、产品列表页面、订单列表页面和修改收货地址页面演示了这几种列表控件的具体使用用法。其中,Spinner 控件主要适用于具有下拉选项的应用场景,ListView 控件常用于数据量较小和简单布局的应用,GridView 主要应用于多行多列的展示,而对于要实现页面布局较复杂或者加载数据量较大的应用场景推荐使用 RecyclerView 控件,这些列表控件在以后的开发中经常使用,需要读者熟练掌握灵活运用。

自测习题

1. BaseAdapter 为什么定义为抽象类? 要想实现自定义的 Adapter,必须实现哪些方法?
2. 简述 SimpleAdapter 对象创建时,各个参数的含义。
3. 根据所学的 ListView 控件,实现“欢乐购商城”的地址列表页面,如图 5-18 所示。
4. 根据所学的 RecyclerView 控件,实现“欢乐购商城”首页中的小图标加载,如图 5-19 所示。



图 5-18 地址列表页面



图 5-19 首页小图标加载