



《移动终端开发技术》 电子教案

第四单元 RecyclerView 的功能和用法

所属专业（教研室）： 计算机软件技术

制定人： 陈媛媛

合作人：

制定时间： 2018年2月

日照职业技术学院



单元标题	RecyclerView 的功能和用法		单元教学学时	4 课时
			在整体设计中的位置	第 8 次
授课班级			上课地点	一体化教室
上课时间	周 月 日第 节			
教学目标	能力目标	知识目标	素质目标	
	能够熟练使用 RecyclerView 制作实现普通布局、横向滚动布局和瀑布流布局效果。	1、要求学生掌握使用 ArrayAdapter 创建 RecyclerView 的方法； 2、要求学生掌握实现横向滚动布局和瀑布流布局效果； 3、要求学生掌握为 RecyclerView 添加点击事件的方法。	1、养成积极主动学习意识； 2、养成勤于动手的习惯。	
教学重点、难点	教学重点：使用 ArrayAdapter 创建 RecyclerView 的方法 教学难点：实现瀑布流布局效果			
教学方法	采用反转课堂教学模式，课前学生学习微课了解知识点，课上采用教师引导、演示，学生分组练习、讨论等教学方法。 运用多媒体、AndroidStudio 开发环境、实训助手、教学平台等辅助授课。			
课前需掌握的知识点	由于 RecyclerView 控件并不在 Andorid SDK 中的，而是在 support v7 包中，因此我们要手动添加该控件。在 app/build.gradle 文件中添加如下依赖： <pre>dependencies { ... compile 'com.android.support:appcompat-v7:23.1.1' //Toolbar compile 'com.android.support:recyclerview-v7:23.1.1' //RecyclerView }</pre>			
教学任务分解	任务一 RecyclerView 的简单用法 任务二 实现横向滚动和瀑布流布局 任务三 实现 RecyclerView 的点击事件			
教学总结				



一、情景导入

在 Android 开发中，我们经常与 ListView、GridView 打交道，它们为数据提供了列表和视图的展示方式，方便用户的操作。然而，随着 Android 的不断发展，单一的 listview 逐渐满足不了需求多变的项目了，因此，谷歌在 support v7 中，加入了新的控件——RecyclerView，该控件整合了 ListView、GridView 的特点，而且最大的优点是可以很方便实现瀑布流效果，因此 RecyclerView 受到越来越多的开发者重视。所以，学习 RecyclerView 的使用也是很有必要的。

二、课前测试

1、RecyclerView 控件整合了 ListView、GridView 的特点，而且最大的优点是可以很方便实现（）效果。

- A. 列表
- B. 瀑布流
- C. 点击
- D. 图文并茂

2、为 RecyclerView 添加适配器的方法为（）。

- A. setOnClickListener() (为 ListView 注册监听器)
- B. setAdapter()
- C. findViewById()
- D. setEmptyView() (给 ListView 设置在空数据下显示的默认提示)

3、调用 LinearLayoutManager 的（）方法可以设置布局的排列方向。

- A. setOrientation()
- B. setAdapter()
- C. findViewById()
- D. getAdapterPosition()

答案 A

4、（）可以用于实现瀑布流布局。

- A. GridLayoutManager//用于实现网络布局
- B. LinearLayoutManager
- C. StaggeredGridLayoutManager
- D. LayoutManager

答案 C

5、RecyclerView 无法实现子项中任意控件或布局的点击事件（）。答案 错

三、知识讲解

任务一 RecyclerView 基本用法

(1) 在 build.gradle 中添加依赖

由于该控件并不在 Andorid SDK 中的，而是在 support v7 包中，因此我们要手动添加该控件。在 app/build.gadle 文件中添加如下依赖：

```
dependencies {  
    ...  
    compile 'com.android.support:appcompat-v7:23.1.1' //Toolbar  
    compile 'com.android.support:recyclerview-v7:23.1.1' //RecyclerView  
}
```

(2) Fruit 类和 fruit_item.xml 和 ListView 中的例子一样。



(3) 创建 FruitAdapter 类

```
FruitAdapter.java
public class FruitAdapter extends RecyclerView.Adapter <FruitAdapter.ViewHolder>{
    private List<Fruit> mFruitList;
    static class ViewHolder extends RecyclerView.ViewHolder{
        ImageView fruitImage;
        TextView fruitName;
        public ViewHolder(View itemView) {
            super(itemView);
            fruitImage=(ImageView) itemView.findViewById(R.id.fruit_image);
            fruitName=(TextView) itemView.findViewById(R.id.fruit_name);
        }
    }
    public FruitAdapter(List<Fruit> fruitList){
        mFruitList=fruitList;
    }
    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view=
        LayoutInflater.from(parent.getContext()).inflate(R.layout.fruit_item,parent,false);
        RecyclerView.ViewHolder holder=new ViewHolder(view);
        return (ViewHolder) holder;
    }
    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {
        Fruit fruit=mFruitList.get(position);
        holder.fruitImage.setImageResource(fruit.getImageId());
        holder.fruitName.setText(fruit.getName());
    }
    @Override
    public int getItemCount() {
        return mFruitList.size();
    }
}
```

(4) 修改 MainActivity.java 代码

```
MainActivity.java
public class MainActivity extends AppCompatActivity {
    private List<Fruit> fruitList=new ArrayList<>();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initFruits();
    }
}
```



```
RecyclerView  
recyclerView=(RecyclerView)findViewById(R.id. recycler_view);  
  
LinearLayoutManager layoutManager=new LinearLayoutManager(this);  
recyclerView.setLayoutManager(layoutManager);  
  
FruitAdapter adapter=new FruitAdapter(fruitList);  
recyclerView.setAdapter(adapter);  
  
}  
  
private void initFruits() {  
    for(int i=0;i<2;i++) {  
        Fruit banana=new Fruit("Banana",R.drawable. banana1);  
        fruitList.add(banana);  
        Fruit cherry=new Fruit("Cherry",R.drawable. cherry1);  
        fruitList.add(cherry);  
        Fruit grape=new Fruit("Grape",R.drawable. grape1);  
        fruitList.add(grape);  
        Fruit mango=new Fruit("Mango",R.drawable. mango1);  
        fruitList.add(mango);  
        Fruit orange=new Fruit("Orange",R.drawable. orange1);  
        fruitList.add(orange);  
        Fruit pear=new Fruit("Pear",R.drawable. pear1);  
        fruitList.add(pear);  
        Fruit pineapple=new Fruit("Pineapple",R.drawable. pineapple1);  
        fruitList.add(pineapple);  
        Fruit tomato=new Fruit("Tomato",R.drawable. tomato1);  
        fruitList.add(tomato);  
    }  
}
```

任务二 实现横向滚动和瀑布流布局

实现横向滚动效果主要代码:

```
LinearLayoutManager layoutManager=new LinearLayoutManager(this);  
layoutManager.setOrientation(LinearLayoutManager.HORIZONTAL);  
recyclerView.setLayoutManager(layoutManager);
```

瀑布流布局效果主要代码:

```
StaggeredGridLayoutManager layoutManager = new StaggeredGridLayoutManager(3,  
StaggeredGridLayoutManager.VERTICAL);
```

案例、实现横向滚动

(1) 修改 fruit_item.xml 文件

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```



```
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <ImageView
        android:id="@+id/fruit_image"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"/>
        <TextView
            android:id="@+id/fruit_name"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:layout_marginLeft="10dp"/>
    </LinearLayout>
```

(2) 修改 MainActivity.java

```
加上: layoutManager.setOrientation(LinearLayoutManager.HORIZONTAL);
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    initFruits();
    RecyclerView recyclerView=(RecyclerView)findViewById(R.id.recycler_view);
    LinearLayoutManager layoutManager=new LinearLayoutManager(this);
    layoutManager.setOrientation(LinearLayoutManager.HORIZONTAL);
    recyclerView.setLayoutManager(layoutManager);
    FruitAdapter adapter=new FruitAdapter(fruitList);
    recyclerView.setAdapter(adapter);
}
```

案例、实现瀑布效果

(1) 修改 fruit_item.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:orientation="vertical">
    <ImageView
    android:id="@+id/fruit_image"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"/>
    <TextView
        android:id="@+id/fruit_name"
        android:layout_width="wrap_content"
```



```
        android:layout_height="wrap_content"
        android:layout_gravity="left"
        android:layout_marginLeft="10dp"/>
</LinearLayout>

(2) 修改 MainActivity.java 代码
//实现瀑布流布局
StaggeredGridLayoutManager layoutManager = new StaggeredGridLayoutManager(3,
StaggeredGridLayoutManager.VERTICAL);
//LinearLayoutManager layoutManager=new LinearLayoutManager(this);
//layoutManager.setOrientation(LinearLayoutManager.HORIZONTAL);
private void initFruits() {
    for (int i = 0; i < 2; i++) {
        Fruit banana = new Fruit(getRandomLengthName("Banana"), R.drawable.banana1);
        fruitList.add(banana);
        Fruit cherry = new Fruit(getRandomLengthName("Cherry"), R.drawable.cherry1);
        fruitList.add(cherry);
        Fruit grape = new Fruit(getRandomLengthName("Grape"), R.drawable.grape1);
        fruitList.add(grape);
        Fruit mango = new Fruit(getRandomLengthName("Mango"), R.drawable.mango1);
        fruitList.add(mango);
        Fruit orange = new Fruit(getRandomLengthName("Orange"), R.drawable.orange1);
        fruitList.add(orange);
        Fruit pear = new Fruit(getRandomLengthName("Pear"), R.drawable.pear1);
        fruitList.add(pear);
        Fruit pineapple = new Fruit(getRandomLengthName("Pineapple"),
R.drawable.pineapple1);
        fruitList.add(pineapple);
        Fruit tomato = new Fruit(getRandomLengthName("Tomato"), R.drawable.tomato1);
        fruitList.add(tomato);
    }
}

private String getRandomLengthName(String name) {
    Random random = new Random();
    int length = random.nextInt(20) + 1;
    StringBuilder builder = new StringBuilder();
    for (int i = 0; i < length; i++)
        builder.append(name);
    return builder.toString();
}
```

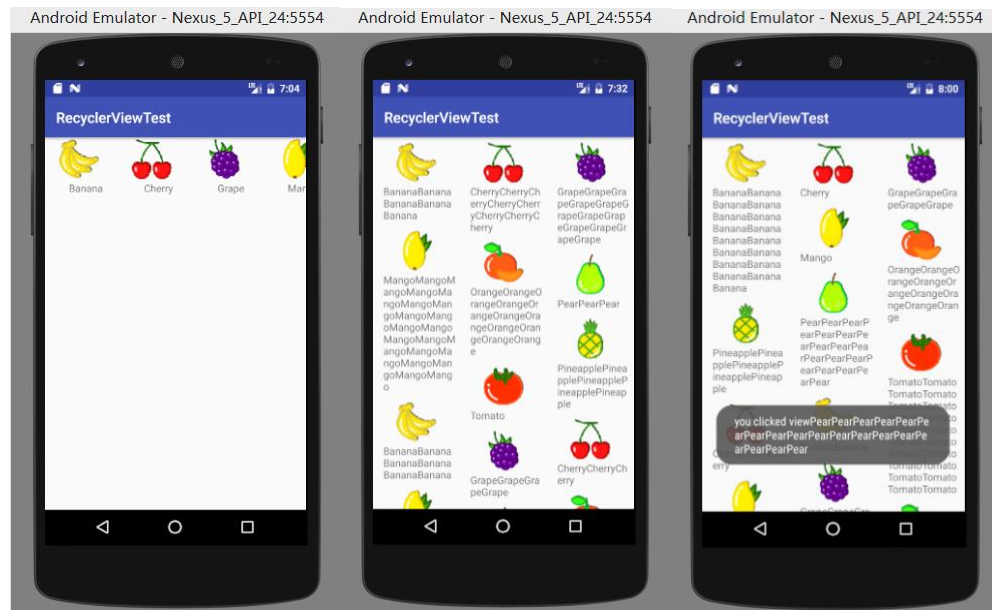
任务三 RecyclerView 的点击事件

修改 FruitAdapter.java 代码

```
static class ViewHolder extends RecyclerView.ViewHolder{
```



```
View fruitView;//点击事件新加
ImageView fruitImage;
TextView fruitName;
public ViewHolder(View itemView) {
    super(itemView);
    fruitView=itemView;//点击事件新加
    fruitImage=(ImageView) itemView.findViewById(R.id. fruit_image);
    fruitName=(TextView) itemView.findViewById(R.id. fruit_name);
}
}
public FruitAdapter(List<Fruit> fruitList) {
    mFruitList=fruitList;
}
@Override
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View view=
    LayoutInflater.from(parent.getContext()).inflate(R.layout. fruit_item,parent, false);
    final ViewHolder holder=new ViewHolder(view);
    holder.fruitView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            int position=holder.getAdapterPosition();
            Fruit fruit=mFruitList.get(position);
            Toast.makeText(view.getContext(), "you clicked
view"+fruit.getName(), Toast.LENGTH_LONG).show();
        }
    });
    holder.fruitImage.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            int position=holder.getAdapterPosition();
            Fruit fruit=mFruitList.get(position);
            Toast.makeText(view.getContext(), "you clicked
view"+fruit.getName(), Toast.LENGTH_LONG).show();
        }
    });
    return holder;
}
```

例 2 效果图

例 3 效果图

例 4 效果图

四、知识巩固

- 1、总结知识点，使用教学平台中的随堂练习题巩固本所学知识。
- 2、使用教学平台中的测试题给学生布置作业。

拓展
作业

点击水果列表，显示水果名称、价格、产地等介绍

教学
后记